

METHODE DOPTIMISATION PAR INTERPOLATION
DE LA FONCTION OBJECTIF
APPLICATION A UN PROBLEME DE CONTROLE OPTIMAL

THESE

pour obtenir le grade de

DOCTEUR - INGENIEUR

MATHEMATIQUES - APPLIQUEES

par

NGUYEN-VAN-TUAN

Ingenieur E.N.S.E.M., Maître es sciences

Thèse soutenue le 15 juin 1973 devant le jury :

MM LEGRAS J. Président

GILORMINI C. Examinateur

LUCIUS M. Examinateur

METHODE D'OPTIMISATION PAR INTERPOLATION
DE LA FONCTION OBJECTIF
APPLICATION A UN PROBLEME DE CONTROLE OPTIMAL

par NGUYEN VAN TUAN



Digitized by the Internet Archive
in 2026

A mes parents, mes frères et mes sœurs

Je voudrais exprimer ma profonde reconnaissance
à Monsieur le Professeur J. LEGRAS qui m'a constamment
guidé dans mon travail avec attention et bienveillance.

Je remercie Messieurs les Professeurs C.
GILORMINI et M. LUCIUS qui me font l'honneur de composer
le Jury.

TABLE DES MATIERES

PREMIERE PARTIE CAS DES VARIABLES LIBRES

I - Méthode de quadratisation par interpolation

I - 1 Rappel de la méthode de quadratisation osculatrice

I - 2 Exposé de la méthode de quadratisation par interpolation

- * Exposé théorique
- * Plan de recherche et organigramme
- * Résultats numériques des essais.

II - Méthode de quadratisation partielle

II - 1 Introduction

II - 2 Proposition

II - 3 Méthode de quadratisation partielle empirique

II - 4 Plan de recherche

II - 5 Technique proposée pour la recherche de $\mathbf{R}^p$

II - 6 Organigramme

II - 7 Résultats numériques des essais

II - 8 Méthode mixte de quadratisation partielle et du gradient

- * Proposition
- * Résultats numériques

II - 9 Temps de calcul de certains essais

III - Conclusion

DEUXIEME PARTIE CAS DES VARIABLES LIEES

I - Introduction et position du problème

II - Exposé de la méthode du gradient projeté

II - 1 Coefficients de Kühn et Tucker

II - 2 Caractérisation d'un maximum local

- * Calcul de α pour une direction $[D]$ quelconque
- * Caractérisation du maximum, conditions de Kühn et Tucker

II - 3 Technique de libération des contraintes saturées au cas où il y a au moins un $\lambda_k > 0$

- III - Plan de recherche du maximum sous contraintes
- IV - Technique de programmation
- V - Organigramme
- VI - Résultats des essais numériques

TROISIEME PARTIE

APPLICATION A LA RESOLUTION D'UN
PROBLEME DE COMMANDE OPTIMALE

- I - Rappel du problème de contrôle optimal
- II - Problème sans contrainte
 - II - 1 Position du problème et sa formulation mathématique
 - II - 2 Méthode de résolution utilisée
 - II - 3 Résultats numériques
- III - Problème avec contraintes
- IV - Conclusion

.....

INTRODUCTION

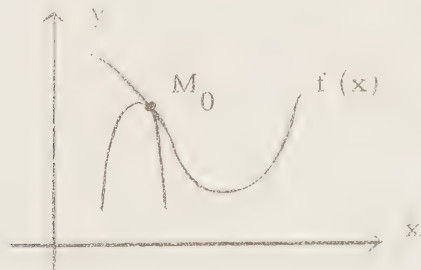
Dans cette étude, le problème posé est de déterminer un optimum local d'une fonction réelle

$$\begin{cases} f([X]) & [X] \in (\Omega) \\ \text{où } (\Omega) \subset \mathbb{R}^n \end{cases}$$

En ce qui concerne le cas sans contrainte ($\Omega = \mathbb{R}^n$), il existe principalement deux familles de méthodes pour le résoudre. La première est celle des méthodes d'approximation du premier ordre dont les principales sont la méthode du gradient et ses variantes. La seconde est celle des méthodes d'approximation du second ordre dans lesquelles le gradient et le hessien sont calculés d'une façon stricte à partir de leur expression analytique.

Ces deux familles présentent deux défauts suivants :

- i) Elles sont difficilement applicables dans certains problèmes tels que ceux du problème de contrôle optimal où l'on ne peut pas expliciter les dérivées.
- ii) Elles deviennent aberrantes dans certains cas quand nous sommes loin de l'optimum. La figure ci-dessous illustre le cas où les renseignements donnés par la parabole osculatrice en M_0 sont complètement inutiles.



Ces inconvénients nous amènent à étudier dans la première partie de ce travail, deux techniques qui sont susceptibles d'améliorer les méthodes de la seconde famille, ce sont les suivantes :

1) Méthode de quadratisation par interpolation .

Il s'agit d'approcher, au point $[X^r]$, la fonction $f([X])$ par un polynôme du second degré par rapport à l'ensemble des variables $(x_1, x_2, \dots, x_n)$, qui interpole $f([X])$ sur un support bien choisi.

2) Méthode de quadratisation partielle

Appelons $[U_n(X^r)]$ la matrice qui désigne soit le hessien $[H_n(X^r)]$, soit un approchant $[A_n(X^r)]$ du hessien suivant que nous sommes dans le cas de quadratisation osculatrice ou par interpolation. L'indice n indique que nous sommes dans l'espace $\mathbb{R}^n$.

Cette matrice $[U_n(X^r)]$ n'est pas en général strictement définie positive (cas de la recherche du minimum), surtout quand nous sommes loin de l'optimum. Ces deux méthodes peuvent donc devenir inefficaces. La méthode de quadratisation partielle consiste à appliquer l'une de ces deux méthodes uniquement aux variables qui appartiennent à un sous-espace $\mathbb{R}^p$ de $\mathbb{R}^n$ pour lequel la matrice $[U_p(X^r)]$ est strictement définie positive.

Nous soulignons que dans les applications que nous ferons de cette technique, nous n'utiliserons que les approchants du gradient et du hessien. Par conséquent, le terme d'erreur des techniques proposées sont mal connu. Il n'est petit qu'au voisinage du support. Il s'agit donc des techniques empiriques.

Ensuite, dans la deuxième partie, nous appliquons la méthode de quadratisation partielle qui sera associée avec une autre technique de montée que nous y exposerons, pour la recherche d'un optimum avec contraintes. Le théorème de Kuhn et Tucker relatif aux conditions d'optimum local sera utilisé pour le test de fin du programme.

Ces techniques (quadratisation par interpolation ou quadratisation partielle) sont bien adaptées à certains problèmes d'optimisation, en particulier, aux problèmes de contrôle optimal ou d'identification. Dans la troisième partie, nous donnerons deux exemples d'application des problèmes de contrôle optimal.

PREMIERE PARTIE

ETUDE DU CAS DES VARIABLES LIBRES

I - METHODE DE QUADRATISATION PAR INTERPOLATION

I - 1 Rappel de la méthode de quadratisation osculatrice

Cette méthode consiste à approcher la fonction $f([X])$ ($[X] \in \mathbb{R}^n$) par son développement de Taylor jusqu'au second ordre. Supposons que nous sommes qu'au point $[X^r]$ (r = indice d'itération), ce développement s'écrit :

$$f([X]) \sim P([X]) = f([X^r]) + [G_n(X^r)]^T [X - X^r] + \frac{1}{2} [X - X^r]^T [H_n(X^r)] [X - X^r] \quad (1.1)$$

où :

$[G_n(X^r)]$ = matrice colonne des dérivées premières au point $[X^r]$

$[X - X^r]$ = matrice transposée de $[X - X^r]$

$[H_n(X^r)]$ = hessien au point $[X^r]$.

Le point $[X']$ où $[P'([X])] = 0$ est défini par :

$$\begin{aligned} [G_n(X^r)] + [X' - X^r] [H_n(X^r)] &= 0 \quad (I, 2) \\ \Rightarrow [X'] &= [X^r] - [H_n(X^r)]^{-1} [G_n(X^r)] \end{aligned}$$

nous définissons une suite $[X^r]$ par la relation (I, 2) :

$$[X^{r+1}] = [X^r] - [H_n(X^r)]^{-1} [G_n(X^r)] \quad (I, 3)$$

nous constatons que si cette suite converge vers un point $[X^*]$, la relation (I, 3) entraîne :

$$[G_n(X^*)] = 0$$

$[X^*]$ est bien un point stationnaire de $f([X])$. Nous rappelons que $[X^*]$ est un point minimum si le hessien $[H_n(X^*)]$ est strictement définie positive.

La formule (I, 3) est bien la formule itérative de la méthode de quadratisation osculatrice.

L'inconvénient de cette méthode est que dans certains problèmes comme les processus physiques ou numériques par exemple, il se peut qu'il soit difficile ou impossible de calculer les dérivées secondes ou même le gradient. Cet inconvénient nous conduit à étudier la méthode suivante que nous appelons "méthode de quadratisation par interpolation".

Exposé de la méthode de quadratisation par interpolation

Etude théorique

Partant d'un point $[X^r]$, au lieu d'approcher la fonction par sa quadratique osculatrice, nous l'approchons par interpolation par une quadratique de la forme :

$$f([X]) = a + [B_n(X^r)]^T [X - X^r] + \frac{1}{2} [X - X^r]^T [A_n(X^r)] [X - X^r] \quad (I, 4)$$

$$\begin{cases} [B_n(X^r)] = \text{matrice colonne } (b_i) \\ [A_n(X^r)] = \text{matrice carrée d'ordre } n \text{ } (a_{ij}) \\ a = \text{scalaire.} \end{cases}$$

Il y a donc au total $N = \frac{(n+1)(n+2)}{2}$ coefficients à calculer. A priori, nous pouvons choisir un support de N points $[X_i]$ quelconques. Puis, il suffit de résoudre le système de N équations à N inconnues

$$f([X_i]) = a + [B_n(X^r)]^T [X_i - X^r] + \frac{1}{2} [X_i - X^r]^T [A_n(X^r)] [X_i - X^r] \quad (I, 5)$$

$i = 1, 2, \dots, N$

afin d'obtenir a , $[B_n(X^r)]$, $[A_n(X^r)]$.

Notons que l'ordre N du système augmente vite quand le nombre n des variables augmente. Pour simplifier la résolution de ce système, nous nous proposons de prendre le support défini comme suit :

- 1°) Point M^r correspondant au tableau $[X^r]$
 - 2°) n points M_{i+}^r correspondant au tableau $[X_i^r]$ $i=1, 2, \dots, n$
- où $[X_{i+}^r] = [x_1^r, x_2^r, \dots, x_{i-1}^r, x_i^r + \ell_i, x_{i+1}^r, \dots, x_n^r]$

3°) n point M_{i-}^r correspondant au tableau $[X_{i-}^r]$ $i = 1, \dots, n$

$$\text{où } [X_{i-}^r] = [x_1^r, x_2^r, \dots, x_{i-1}^r, x_i^r - \ell_i, x_{i+1}^r, \dots, x_n^r]$$

4°) $\frac{(n-1)(n-2)}{2}$ point M_{ij}^r correspondant au tableau $[X_{ij}^r]$

$$\text{où } [X_{ij}^r] = [x_1^r, \dots, x_{i-1}^r, x_i^r + \ell_i, x_{i+1}^r, \dots, x_{j-1}^r, x_j^r + \ell_j, x_{j+1}^r, \dots, x_n^r]$$

$$i = 1, 2, \dots, n$$

$$j = i + 1, \dots, n$$

Les figures (1), (2) illustrent le support d'interpolation ainsi choisi dans les cas 2 et 3 variables.

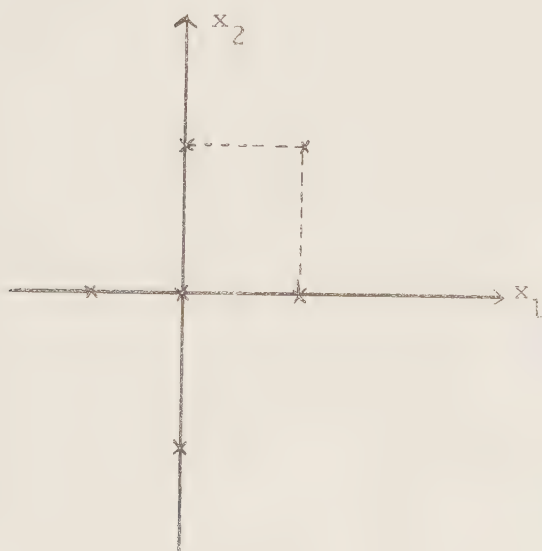


figure (1)

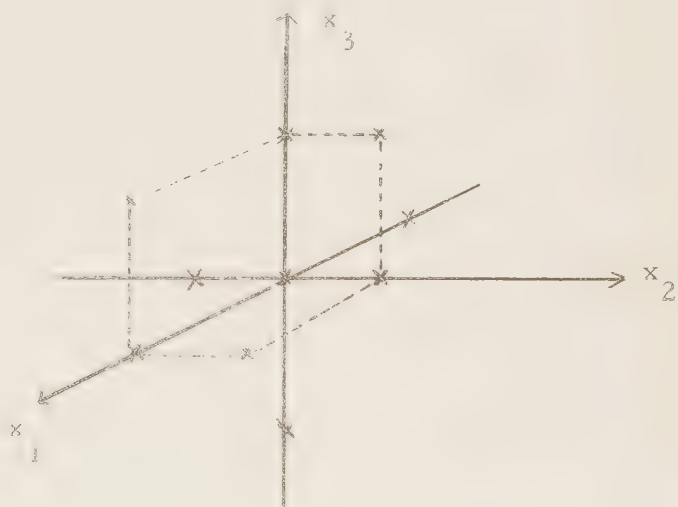


figure (2)

En écrivant le système (5) correspondant à ce support, nous obtenons :

- pour le point $[X^r] \Rightarrow a = f(X^r)$ (I, 6a)

- pour les points $[X_{i+}^r]$ et $[X_{i-}^r]$:

$$\begin{cases} a + b_i \ell_i + \frac{1}{2} a_{i,i} \ell_i^2 = f(X_{i+}^r) \\ a - b_i \ell_i + \frac{1}{2} a_{i,i} \ell_i^2 = f(X_{i-}^r) \end{cases} \quad i = 1, 2, \dots, n$$

$$\text{d'où : } \begin{cases} b_i = \frac{f(X_{i+}^r) - f(X_{i-}^r)}{2 \ell_i} & (I, 6b) \\ a_{i,i} = \frac{f(X_{i+}^r) + f(X_{i-}^r) - 2 f(X^r)}{\ell_i^2} & (I, 6c) \end{cases}$$

Pour un point $[X_{ij}^r]$:

$$a + b_i \ell_i + b_j \ell_j + \frac{1}{2} [a_{ii} \ell_i^2 + 2 a_{ij} \ell_i \ell_j + a_{jj} \ell_j^2] = f(X_{ij}^r)$$

$$i = 1, 2, \dots, n$$

$$j = i + 1, \dots, n$$

En tenant compte des relations (6a), (6b), (6c), nous avons :

$$a_{ij} = \frac{f(X_{ij}^r) - f(X_{i+}^r) - f(X_{j+}^r) + f(X^r)}{\ell_i \ell_j} \quad (I, 6d)$$

Nous constatons qu'avec ce support d'interpolation, la méthode consiste à remplacer les dérivées de $f(X)$ par leur approchant :

$$\begin{aligned} f'_{x_i} &\longrightarrow b_i \\ f''_{x_i^2} &\longrightarrow a_{ii} \\ f''_{x_i x_j} &\longrightarrow a_{ij} \\ &\quad i \neq j \end{aligned}$$

Puis nous effectuons les calculs comme précédemment, nous obtenons la formule itérative de cette méthode :

$$[X^{r+1}] = [X^r] - [A_n(X^r)]^{-1} [B_n(X^r)] \quad (I, 7)$$

Remarques

- 1) Lorsque les pas ℓ_i, ℓ_j deviennent très petits, les matrices $[A_n(X^r)]$ et $[B_n(X^r)]$ sont des bonnes approximations du Hessien $[H_n(X^r)]$ et du gradient $[G_n(X^r)]$. Dans ce cas, il n'y a pas de différence essentielle entre les deux méthodes à condition que $f([X])$ puisse être déterminée.

avec une assez grande précision (peu d'erreur de chute et peu d'erreur expérimentale) pour que le calcul des b_i , a_{ij} , a soit satisfaisant.

- 2) La technique de quadratisation osculatrice n'est sûre qu'au voisinage de la solution, donc il va de même pour la méthode de quadratisation par interpolation avec la condition supplémentaire que les pas soient suffisamment petits.
- 3) Le choix particulier du support nous a permis d'explicitier les éléments des matrices $[A_n(X^r)]$ et $[B_n(X^r)]$. Par contre, si, par exemple par suite des contraintes expérimentales, nous ne pouvons pas fixer un support comme nous l'avons défini, nous pouvons utiliser tout support quelconque de $\frac{(n+1)(n+2)}{2}$ points. Puis pour calculer $[A_n(X^r)]$ et $[B_n(X^r)]$, il suffit de résoudre le système linéaire (I, 5). La méthode a été testée pour 6 variables, et nous n'avons pas rencontré d'instabilité numérique.

I.2.2 Plan de recherche et organigramme

C'est une méthode itérative. Chaque itération comprend les trois phases principales suivantes :

Supposons que nous sommes au point $[X^r]$:

1ère phase Calcul des matrices $[A_n(X^r)]$ et $[B_n(X^r)]$ par les relations (I, 6) $[(I, 6b), (I, 6c), (I, 6d)]$.

2ème phase Calcul du nouveau point $[X^{r+1}]$ en résolvant le système linéaire :

$$[A_n(X^r)] [X^{r+1} - X^r] + [B_n(X^r)] = 0 \quad (I, 8)$$

3ème phase Test d'arrêt.

La première phase est évidente. Il suffit d'utiliser les relations (I, 6) en faisant appel au sous-programme calculant la valeur de la fonction en un point.

La seconde phase n'est possible que si le système (I, 8) est soluble. Nous avons utilisé la procédure existante MCADIR pour la résolution de ce système. Cette procédure s'appuie sur les méthodes de résolution d'un système surdimensionné décrites dans la référence [IV]. Dans ce sous programme, il y a un indicateur IP (1) défini comme suit :

* $IP(1) = 0 \iff$ système soluble

* $IP(1) \neq 0 \iff$ sinon.

Donc il suffit de placer un test sur $IP(1)$ à la sortie de ce sous-programme :

* si $IP(1) = 0$ nous commençons la troisième phase.

* sinon, c'est-à-dire si le système $(I, 8)$ n'est pas soluble par la procédure MCADIR. Les pas sont donc mal choisis. Nous recommençons la première phase avec des nouveaux pas obtenus en divisant les anciens par 2. Nous arrêtons le programme quand le pas devient très petit et le système reste toujours insoluble.

Appelons $FM0 = f([X^r])$ et $FM1 = f([X^{r+1}])$ et plaçons nous dans le cas de la recherche du minimum.

Nous commençons les tests en comparant $FM0$ et $FM1$. S'il n'y a pas d'amélioration, c'est-à-dire si $FM1 \geq FM0$, alors nous recommençons la première phase à partir de X^r en divisant les pas par deux.

Sinon, c'est-à-dire si $FM1 < FM0$, nous testons sur :

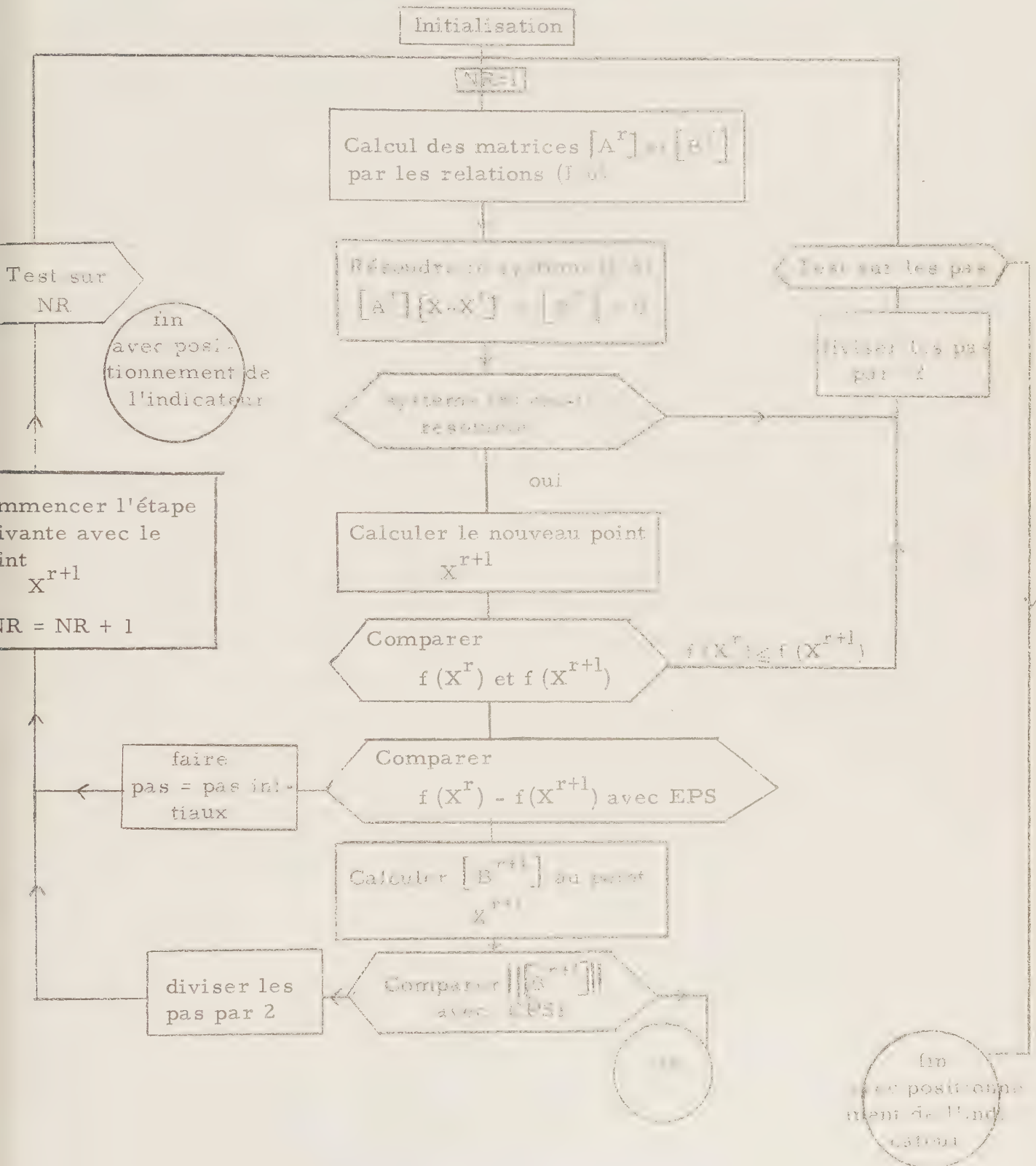
$$\Delta f = FM0 - FM1.$$

Si $\Delta f \leq \varepsilon$ où ε est une valeur prise comme paramètre nous admettons que nous sommes très près de la solution, alors le programme se continue par un test portant sur le gradient de la forme quadratique interpolante, c'est-à-dire sur les coefficients b_i au point $[X^{r+1}]$. Si tous les $|b_i| \leq \varepsilon_1$, la recherche est terminée. Sinon, nous recommençons la première phase avec $[X^r] = [X^{r+1}]$ en divisant les pas par 2.

Si $\Delta f > EPS$, nous recommençons la première phase avec $[X^r] = [X^{r+1}]$ en donnant aux pas leur valeur initiale.

Le début de chaque étape est caractérisé par une amélioration de la valeur de la fonction. Le nombre d'étapes est limité par un nombre NRMAX pris comme paramètre.

Organigramme de la méthode de quadratisation par interpolation



I.2.3 Résultats numériques des essais

Tous les calculs sont faits en double précision sur C.I.I. 10070.

I.2.3.1 Essai sur

$$f(x_1, x_2) = \left(\frac{x_1^4}{12} + \frac{x_1^2}{2} + \frac{x_2^2}{2} + x_2 + x_1 x_2 \right)$$

$$\begin{cases} f'_{x_1} = \frac{x_1^3}{3} + x_1 + x_2 \\ f'_{x_2} = x_2 + 1 + x_1 \end{cases} \Rightarrow \text{coordonnées du } \begin{cases} x_1 = 1,442 \\ x_2 = -2,442 \end{cases} \text{ pt minimum}$$

le hessien en un point quelconque :

$$[H] = \begin{bmatrix} x_1^2 + 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Un essai est fait le point initial ($x_1 = 0$; $x_2 = -2$) et le pas initial (0.25 ; 0.25). Nous remarquons qu'en ce point initial, le hessien est singulier. Par conséquent, si nous appliquons la méthode de quadratisation osculatrice, le système (I, 2) sera insoluble, nous n'aurons donc pas la solution, par contre la méthode de quadratisation par interpolation donne le résultat figurant dans le tableau ci-dessous :

itér	x_1	x_2	$f([x])$
0	0	-2	0
1	1	-1	-0.917
2	1.539	-2.539	-1.531
3	1.415	-2.415	-1.584 ₍₁₀₇₎
4	1.449	-2.449	-1.582 ₍₁₀₇₎
5	1.44	-2.44	-1.562 ₍₁₀₇₎
6	1.442	-2.442	-1.582 ₍₁₀₇₎

I.2.3.2 Essais sur :

$$f(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2 + e^{x_1^2 + x_2^2} + 1$$

qui est symétrique par rapport à x_1, x_2 .

Les coordonnées du point minimum : (0;0).

1°) Point initial (0.5 ; 0.5)

Pas initial (0.5 ; 0.5)

NR	$x_1 = x_2$	$f([X])$
0	0.5	$0.1399 \cdot 10^1$
1	0.284	0.4165
2	$0.814 \cdot 10^{-1}$	$0.3323 \cdot 10^{-1}$
3	$0.488 \cdot 10^{-2}$	$0.1193 \cdot 10^{-3}$
4	$0.360 \cdot 10^{-4}$	$0.6493 \cdot 10^{-8}$
5	$0.174 \cdot 10^{-6}$	$0.1505 \cdot 10^{-12}$

2°) Point initial (0.5 ; 0.5)

Pas initial (1 ; 1)

NR	$x_1 = x_2$	$f([X])$
1	0.416	0.9341
2	0.319	0.5309
3	0.210	0.2248
4	0.104	$0.5430 \cdot 10^{-1}$
5	$0.311 \cdot 10^{-1}$	$0.4831 \cdot 10^{-2}$
6	$0.493 \cdot 10^{-2}$	$0.1214 \cdot 10^{-3}$
7	$0.543 \cdot 10^{-3}$	$0.1476 \cdot 10^{-5}$
8	$0.557 \cdot 10^{-4}$	$0.1549 \cdot 10^{-7}$
9	$0.565 \cdot 10^{-5}$	$0.1597 \cdot 10^{-9}$
10	$0.271 \cdot 10^{-7}$	$0.3539 \cdot 10^{-14}$

Dans ces deux essais, il est normal que nous obtenons des valeurs égales de x_1 et x_2 à cause du fait que nous partons des conditions initiales qui sont symétriques par rapport à x_1 et x_2 .

3°) Point initial (2 ; 3)

Pas initial (0.5 ; 0.25)

le but de cet essai est de rompre la symétrie des calculs.

NR	x_1	x_2	$f(x)$
0	2	3	$0.4424 \cdot 10^6$
1	1.970	2.967	$0.3237 \cdot 10^6$
2	1.940	2.933	$0.2360 \cdot 10^6$
3	1.904	2.899	$0.1713 \cdot 10^6$
4	1.877	2.864	$0.1239 \cdot 10^6$
5	1.843	2.828	$0.8926 \cdot 10^5$
7	1.772	2.754	$0.4568 \cdot 10^5$
9	1.695	2.677	$0.2295 \cdot 10^5$
12	1.565	2.553	$0.7870 \cdot 10^4$
19	1.291	2.274	$0.7985 \cdot 10^3$
24	0.594	1.960	$0.7080 \cdot 10^2$
30	0.165	1.167	$0.6191 \cdot 10^1$
32	0.350	0.291	0.5393
34	$0.542 \cdot 10^{-2}$	$0.341 \cdot 10^{-2}$	$0.1005 \cdot 10^{-3}$
36	$0.372 \cdot 10^{-4}$	$-0.258 \cdot 10^{-4}$	$0.3143 \cdot 10^{-8}$
37	$0.969 \cdot 10^{-6}$	$0.652 \cdot 10^{-6}$	$0.2096 \cdot 10^{-11}$
38	$0.612 \cdot 10^{-8}$	$-0.409 \cdot 10^{-8}$	$0.2911 \cdot 10^{-16}$

I.2.3.3 Essais sur fonction de 3 variables définies par :

$$f(x_1, x_2, x_3) = \sum_{i=1}^2 \left[(e^{-C_i x_1} - e^{-C_i x_2}) - x_3 (e^{-C_i} - e^{-10C_i}) \right]^2$$

avec $\begin{cases} C_1 = 0.1 \\ C_2 = 1 \end{cases} \quad [\text{réf (III)}]$

Dans la référence (II), Monsieur CHAU-NEARKASEN a démontré que cette fonction admet une infinité de minimums qui sont définis comme suit :

$$\sum_{k=0}^g y_1^k y_2^{g-k} = (e^{-1} - e^{-10}) / (e^{-0.1} - e^{-1})$$
$$\begin{cases} y_1 = e^{-0.1 x_{1m}} \\ y_2 = e^{-0.1 x_{2m}} \end{cases}$$
$$\Rightarrow x_{3m} = \frac{e^{-x_{1m}} - e^{-x_{2m}}}{e^{-1} - e^{-10}}$$

Il y a une infinité de (y_1, y_2) qui vérifie la première relation, donc il y a une infinité de (x_{1m}, x_{2m}, x_{3m}) .

Nous avons fait 2 essais suivants :

1°) Point initial (1.9 ; 1)

Pas initial (0.25 ; 0.25 ; 0.25)

NR	x_1	x_2	x_3	$f(X)$
0	1	9	1	$0.1495 \cdot 10^{-2}$
1	1.114	8.935	0.902	$0.1596 \cdot 10^{-4}$
2	1.168	8.318	0.848	$0.1588 \cdot 10^{-5}$
3	1.222	7.895	0.803	$0.1124 \cdot 10^{-5}$
5	1.275	7.488	0.759	$0.6206 \cdot 10^{-7}$
7	1.295	7.347	0.743	$0.2394 \cdot 10^{-9}$
9	1.300	7.313	0.739	$0.9035 \cdot 10^{-12}$

2°) Point initial (1 ; 9 ; 1)

Pas initial (1 ; 0.25 ; 0.5)

k	x_1	x_2	x_3	$f(X)$
1	1.882	8.283	0.829	$0.2023 \cdot 10^{-3}$
2	1.879	6.411	0.641	$0.1522 \cdot 10^{-3}$
3	1.717	5.161	0.502	$0.8065 \cdot 10^{-4}$
4	1.557	4.824	0.383	$0.6116 \cdot 10^{-4}$
10	1.450	3.864	$0.267 \cdot 10^{-3}$	$0.4726 \cdot 10^{-5}$
15	1.377	4.477	$0.218 \cdot 10^{-5}$	$0.6025 \cdot 10^{-10}$
16	1.377	4.491	$0.682 \cdot 10^{-6}$	$0.3647 \cdot 10^{-11}$
17	1.377	4.497	$0.156 \cdot 10^{-6}$	$0.2247 \cdot 10^{-12}$

Dans ces deux essais, partons d'un même point initial, avec des valeurs différentes des pas h_1 et h_3 , nous constatons que le nombre d'étapes est modifié assez nettement.

I.2.3.4 Essais sur :

$$f(x_1, x_2, x_3, x_4) = x_1^2 + 10 x_2^2 + x_3^2 + 5 x_4^2 + e^{x_2 x_3} - 1$$

les coordonnées du point minimum : (0 ; 0 ; 0 ; 0)

1°) Point initial (1 ; 1 ; 1 ; 1)

Pas initial (0.25 ; 0.25 ; 0.25 ; 0.25)

k	x_1	x_2	x_3	x_4	$f(X)$
0	1	1	1	1	0.1004
1	$0.289 \cdot 10^{-14}$	-0.666	$0.286 \cdot 10^1$	$0.222 \cdot 10^{-15}$	0.1118
2	$0.412 \cdot 10^{-16}$	$-0.761 \cdot 10^{-1}$	0.173 "	$-0.233 \cdot 10^{-15}$	0.2073
3	$0.267 \cdot 10^{-15}$	$0.400 \cdot 10^{-1}$	$0.710 \cdot 10^{-2}$	$0.114 \cdot 10^{-17}$	0.2160
4	$0.344 \cdot 10^{-17}$	$0.291 \cdot 10^{-2}$	$0.641 \cdot 10^{-3}$	$0.117 \cdot 10^{-17}$	0.2336
5	$0.344 \cdot 10^{-17}$	$0.165 \cdot 10^{-2}$	$0.344 \cdot 10^{-3}$	$0.117 \cdot 10^{-17}$	0.2725

2°) Point initial (1 ; 1 ; 1 ; 1)

Pas initial (0.5 ; 0.5 ; 0.5 ; 0.5)

NR	x_1	x_2	x_3	x_4	$f([X])$
0	1	1	1	1	$0.1872 \cdot 10^2$
1	$0.289 \cdot 10^{-14}$	-0.666	$0.286 \cdot 10^1$	$0.222 \cdot 10^{-15}$	$0.1178 \cdot 10^2$
2	$0.274 \cdot 10^{-14}$	$-0.758 \cdot 10^{-1}$	0.150	$-0.120 \cdot 10^{-15}$	$0.6858 \cdot 10^{-1}$
3	$0.414 \cdot 10^{-15}$	$0.153 \cdot 10^{-2}$	$-0.670 \cdot 10^{-2}$	$-0.245 \cdot 10^{-16}$	$0.5804 \cdot 10^{-4}$
4	$0.173 \cdot 10^{-16}$	$-0.520 \cdot 10^{-4}$	$0.131 \cdot 10^{-3}$	$0.273 \cdot 10^{-17}$	$0.3750 \cdot 10^{-7}$
5	$0.343 \cdot 10^{-17}$	$0.113 \cdot 10^{-5}$	$-0.419 \cdot 10^{-5}$	$0.274 \cdot 10^{-17}$	$0.2559 \cdot 10^{-10}$
6	$-0.131 \cdot 10^{-20}$	$-0.782 \cdot 10^{-8}$	$0.221 \cdot 10^{-7}$	$-0.441 \cdot 10^{-22}$	$0.8774 \cdot 10^{-15}$

Nous remarquons qu'avec le même point initial, le pas est doublé,
le nombre d'étape est peu modifié.

3°) Point initial (-3 ; 1 ; 1.5 ; 2.5)

Pas initial (0.25 ; 0.25 ; 0.25 ; 0.25)

NR	x_1	x_2	x_3	x_4	$f(X)$
0	-3	1	-1.5	2.5	$0.5598 \cdot 10^2$
1	$-0.533 \cdot 10^{-14}$	$0.132 \cdot 10$	-0.545	$0.444 \cdot 10^{-15}$	$0.1727 \cdot 10^2$
2	$-0.698 \cdot 10^{-14}$	$0.199 \cdot 10^{-1}$	-0.292	$0.785 \cdot 10^{-14}$	$0.8359 \cdot 10^{-1}$
3	$-0.391 \cdot 10^{-16}$	$0.738 \cdot 10^{-3}$	$-0.821 \cdot 10^{-3}$	$0.614 \cdot 10^{-17}$	$0.5509 \cdot 10^{-5}$
4	$-0.139 \cdot 10^{-16}$	$-0.197 \cdot 10^{-5}$	$0.129 \cdot 10^{-4}$	$0.614 \cdot 10^{-17}$	$0.1768 \cdot 10^{-9}$
5	$-0.139 \cdot 10^{-16}$	$0.226 \cdot 10^{-7}$	$-0.431 \cdot 10^{-7}$	$0.614 \cdot 10^{-17}$	$0.5857 \cdot 10^{-14}$

En comparant avec le premier essai, avec le point initial différent,
le même pas, le nombre d'étapes reste le même.

1.2.3.5 Essais sur 1 fonction de 6 variables définies

comme suit :

Considérons l'équation différentielle :

$$\left\{ \begin{array}{l} \frac{dz}{dy} - z^2 = 1 \\ z(0) = 0 \end{array} \right. \quad [\text{réf (III)}]$$

Nous résolvons cette équation en approchant z par un polynôme de degré $m = 5$ dans $0 \leq y \leq 1$ en minimisant la somme des carrés des résidus aux points sélectionnés

$$R = \sum_{i=1}^{29} \left\{ \sum_{j=2}^{m+1} (0.1x_i y_j^{(j)} - \frac{1}{2} y_j^{(j)2} - 1) \right\}^2$$

Le problème est de trouver $\vec{X}(x_1, x_2, x_3, x_4, x_5, x_6)$ qui minimise R . La solution est :

$$\vec{X}_m (-0.016 ; 1.012 ; -0.233 ; 1.260 ; -1.513 ; 0.993)$$

$$f(\vec{X}_m) = 2.288.10$$

Nous signalons que cette approximation polynômiale n'est pas appropriée en vue de résoudre cette équation différentielle, mais ce qui nous importe est le problème de minimisation de la quantité R .

Nous avons fait des essais suivants :

1) Point initial (0.1 ; 1.5 ; 0.0 ; 1.5 ; -1.0 ; 0.5)

Pas initial (0.5 ; 0.5 ; 0.5 ; 0.5 ; 0.5 ; 0.5)

2) Point initial (0.5 ; 1.0 ; 0.0 ; 1.0 ; -1.3 ; 1.0)

Pas initial (0.25 ; 0.25 ; 0.25 ; 0.25 ; 0.25 ; 0.25)

3) Point initial (0.1 ; 1.5 ; 0.0 ; 1.5 ; -1.0 ; 0.5)

Pas initial (0.25 ; 0.25 ; 0.25 ; 0.25 ; 0.25 ; 0.25)

Les résultats sont respectivement dans les tableaux n° 1, 2, 3. Dans le tableau n° 1 bis, nous décomposons quelques premières étapes du tableau n° 1.

Tableau n° 1 pas initial = 0.5

NR	x_1	x_2	x_3	x_4	x_5	x_6	$f([X])$
0	0.1000	1.5000	0.0000	1.5000	-1.0000	0.5000	$0.3520 \times 10^{+2}$
1	-0.2108	1.3293	0.0073	1.6769	-0.6783	0.7201	$0.2301 \times 10^{+2}$
2	0.5559	1.2838	-1.1171	0.9710	-0.3312	1.8186	0.1217×10^2
3	0.0308	1.5043	-0.7363	1.4786	-0.0623	-0.0719	0.7638×10
4	-0.0690	1.4533	-0.7531	1.4945	-0.1955	0.3568	0.4953×10
5	-0.0570	1.4950	-0.7577	1.4233	-0.8576	0.5971	0.2147×10
10	-0.1851	1.0976	-0.4634	1.0658	-0.8599	0.4419	0.5776×10^{-1}
12	-0.1351	1.0447	-0.4818	1.0664	-0.8484	0.4683	0.2822×10^{-1}
17	-0.0662	1.0276	-0.2687	0.9296	-0.8720	0.5972	0.8334×10^{-2}
31	-0.0281	0.9998	-0.0977	0.7951	-0.9493	0.7442	0.3171×10^{-2}
40	-0.0244	1.0035	-0.1381	0.9349	-1.1192	0.8199	0.2717×10^{-2}
45	-0.0216	1.0067	-0.1727	1.0516	-1.2597	0.8802	0.2465×10^{-2}
52	-0.0161	1.0121	-0.2290	1.2467	-1.4970	0.9856	0.2288×10^{-2}
55	-0.0157	1.0124	-0.2330	1.2604	-1.5137	0.9930	0.2288×10^{-2}

Tableau n° 1 bis

NR	NPAS	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	f([x])
0	1	0.1000	1.5000	0.0000	1.5000	-1.0000	0.5000	0.3520 × 10 ²
1	1	-0.2108	1.3293	0.0073	1.6769	-0.6783	-0.7201	0.2301 × 10 ²
2	1	-0.3316	1.3666	0.1164	1.7946	-0.6700	-0.3330	0.2418 × 10 ²
	2	-0.7786	1.3743	0.4249	2.2875	-0.0732	-2.0421	0.6628 × 10 ²
	3	-0.4558E+02	-0.3008	0.3467E+02	0.5605E+02	0.6597E+02	-0.1856E+03	0.1190 10 ⁹
	4	0.9261	2.0818	-1.1486	-0.5050	-3.6690	6.7914	0.9896 10 ²
	5	0.7564	0.9726	-0.9185	0.9824	-0.8852	2.4501	0.2760 10 ²
	6	0.5559	1.2838	-1.1170	0.9710	-0.3312	1.8186	0.1217 10 ²
3	1	0.6451	1.0537	-1.4233	0.8416	-0.1753	2.8118	0.2996 10 ²
	2	2.5881	-0.0979	-3.4284	-1.0105	-0.5904	9.4487	0.1275 10 ⁴
	3	0.0308	1.5043	-0.7363	1.4786	-0.0623	-0.0711	0.7638 10
4	1	-0.0690	1.4533	-0.7331	1.4945	-0.1955	0.3568	0.4953 10

NR	x_1	x_2	x_3	x_4	x_5	x_6	$f([x])$
0	0.5	1	0	1	-1.3	1	0.2967×10^2
1	0.1988	0.8430	0.1993	1.5942	-2.0204	1.6187	0.1594×10
2	0.2480	0.7457	0.1401	1.5656	-2.1001	1.7447	0.1137×10
3	0.1279	1.2459	-0.2151	0.7398	-1.0807	1.1403	0.3612
4	0.0904	0.9390	-0.0807	1.5617	-2.0551	1.3735	0.1326
5	0.0405	0.9840	0.0408	1.2490	-1.9981	1.4430	0.3239×10^{-1}
7	0.0478	0.9497	0.0136	1.2988	-2.0082	1.4236	0.1973×10^{-1}
9	-0.0208	1.0539	-0.4413	1.7743	-2.0871	1.2197	0.6662×10^{-2}
11	-0.0106	1.0330	-0.3989	1.7541	-2.0836	1.2283	0.3384×10^{-2}
13	-0.0082	1.0270	-0.3822	1.7390	-2.0787	1.2315	0.3192×10^{-2}
15	-0.0039	1.0253	-0.3660	1.7144	-2.0630	1.2357	0.3095×10^{-2}
17	-0.0040	1.0235	-0.3489	1.6665	-2.0108	1.2160	0.2948×10^{-2}
19	-0.0062	1.0222	-0.3351	1.6156	-1.9477	1.1862	0.2792×10^{-2}
21	-0.0095	1.0184	-0.2959	1.4796	-1.7813	1.1124	0.2481×10^{-2}
27	-0.0100	1.0181	-0.2927	1.4675	-1.7660	1.1105	0.2459×10^{-2}
30	-0.0120	1.0161	-0.2719	1.3952	-1.6780	1.0662	0.2361×10^{-2}
33	-0.0154	1.0128	-0.2369	1.2740	-1.5302	1.0003	0.2288×10^{-2}
37	-0.01572	1.01244	-0.23200	1.26045	-1.51375	0.99301	0.2288×10^{-2}

Tableau n° 3 pas initial = 0,25

NR	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	0.10000	1.50000	0.00000	1.50000	-1.00000	0.50000	0.3520,00
2	0.19107	0.87712	-0.19884	1.62324	-1.21212	0.73594	0.2007 × 10
3	0.12642	0.71993	-0.17536	2.03273	-1.51670	0.77693	0.1782 × 10
4	-0.40675	1.35471	-0.05275	-0.08327	-0.78243	1.12408	0.1165 × 10
5	-0.11849	1.02894	-0.38811	1.08409	-0.84890	0.32495	0.7284 × 10
9	-0.03279	0.96749	-0.10694	0.79446	-0.79158	0.59459	0.2177 × 10 ⁻¹
10	-0.06850	1.03626	-0.21657	0.80379	-0.79387	0.59111	0.8823 × 10 ⁻²
12	-0.07451	1.01638	-0.18296	0.78225	-0.80554	0.59308	0.6855 × 10 ⁻²
17	-0.04321	1.00374	-0.11441	0.73344	-0.81396	0.65130	0.4035 × 10 ⁻²
31	-0.02725	1.00047	-0.10581	0.82453	-0.98601	0.76065	0.3059 × 10 ⁻²
43	-0.0202133	1.00678	-0.17309	1.05385	-1.26286	0.88198	0.2460 × 10 ⁻²
49	-0.01611	1.012050	-0.22894	1.24640	-1.49666	0.98541	0.2288 × 10 ⁻²
52	-0.01573	1.01243	-0.23299	1.26041	-1.51371	0.99299	0.2288 × 10 ⁻²

II - METHODE DE QUADRATISATION PARTIELLE

I - 1 Introduction

Considérons les méthodes de quadratisation exposées précédemment :

$$\begin{bmatrix} X^{r+1} \end{bmatrix} = \begin{bmatrix} X^r \end{bmatrix} - \begin{bmatrix} U_n (X^r) \end{bmatrix}^{-1} \begin{bmatrix} V_n (X^r) \end{bmatrix}$$

où $\begin{bmatrix} U_n (X^r) \end{bmatrix}$ et $\begin{bmatrix} V_n (X^r) \end{bmatrix}$ représentent respectivement le hessien $\begin{bmatrix} H_n (X^r) \end{bmatrix}$ et le gradient $\begin{bmatrix} G_n (X^r) \end{bmatrix}$ (cas de quadratisation osculatrice) ou leur approchant $\begin{bmatrix} A_n (X^r) \end{bmatrix}$ et $\begin{bmatrix} B_n (X^r) \end{bmatrix}$ calculés par les formules (I, 6) (cas de quadratisation par interpolation).

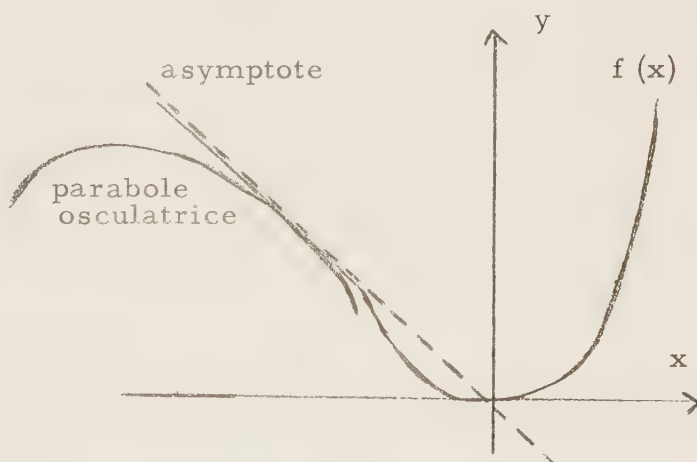
Ces méthodes sont en général excellentes au voisinage de l'optimum. Par contre, quand nous en sommes loin, elles peuvent devenir aberrantes. En effet, prenons l'exemple du cas simple suivant :

$$f(x) = xe^x - x$$

point minimum $x = 0$

Supposons que nous sommes au point $x^r = -3$. Alors avec la méthode de quadratisation osculatrice :

$$x^{r+1} \sim -3 - e^3$$



Nous obtenons donc des points qui deviennent de plus en plus éloignés du point minimum.

Dans le cas de plusieurs variables, si nous tenons compte de tous les termes du second degré, à chaque itération, nous n'avons pas forcément une amélioration, autrement dit, ces méthodes ne sont pas des méthodes de descente certaine, (ou de montée certaine). Afin de surmonter ce point faible de ces méthodes, nous nous proposons de les modifier pour les transformer en technique de descente certaine dont l'idée de base est la suivante :

"En chaque point $[X^r]$, au lieu d'approcher par une quadratique dans l'espace $\mathbb{R}^n$ dont la matrice $[U_n(X^r)]$ est quelconque, nous nous proposons de chercher un sous-espace $\mathbb{R}^p$ de $\mathbb{R}^n$ tel que la matrice $[U_p(X^r)]$ correspondant est strictement défini positive".

Nous démontrerons que la direction $[U_p(X^r)]^{-1} [V_p(X^r)]$ est une direction de descente dans le cas de quadratisation osculatrice. Dans le second paragraphe, nous approcherons $f([X])$ par interpolation, nous adapterons la technique décrite et exposerons la technique empirique correspondante.

II - 2 Proposition

Soit la fonction $f([X]), [X] \in \mathbb{R}^p$.

Considérons son développement de Taylor jusqu'au second ordre au voisinage de $[X^r]$:

$$f([X]) \simeq f([X^r]) + [G_p(X^r)]^T [X - X^r] + \frac{1}{2} [X - X^r]^T [H_p(X^r)] [X - X^r] + 0 \|X - X^r\|^3 \quad (I, 9)$$

$$\left\{ \begin{array}{l} \text{Si } [H_p(X^r)] \text{ est strictement définie positive et si } [X^*] \text{ est solution} \\ \text{de l'équation :} \\ [G_p(X^r)] + [H_p(X^r)] [X - X^r] = 0 \end{array} \right. \quad (I, 10)$$

Alors la direction $[X^* - X^r]$ est une direction de descente.

Considérons le segment $M^* M^r$ défini par :

$$[X] = [X^r] + t [X^* - X^r] \text{ avec } 0 \leq t \leq 1 \quad (I, 11)$$

$$\Rightarrow [X - X^r] = t [X^* - X^r] \quad (I, 12)$$

En remarquant que $[H_p(X^r)] = [H_p(X^r)]^T = [H_p(X^r)]^T$ et en utilisant les relations (I, 10), (I, 11), (I, 12), la relation (I, 9) devient :

$$f([X]) - f([X^r]) = - \left(t - \frac{t^2}{2}\right) [X^* - X^r]^T [H_p(X^r)] [X^* - X^r] + 0 + t \|X^* - X^r\|^3$$

$$\text{Avec la condition } 0 \leq t \leq 1 \Rightarrow t - \frac{t^2}{2} > 0. \quad (I, 13)$$

Donc avec l'hypothèse : " $[H_p(X^r)]$ strictement définie positive", le premier terme est négatif. Pour des valeurs de t suffisamment petit, le terme

d'erreur $0 \parallel t [X^* - X^r] \parallel^3$ est négligeable devant ce premier, nous avons alors :

$$f([X]) - f([X^r]) < 0 \Rightarrow f([X]) < f([X^r])$$

Remarque : Dans le cas où $f([X])$ est une forme quadratique, il n'y a pas de terme d'erreur. La quantité $f([X]) - f([X^r])$ est minimum si $(t - \frac{t^2}{2})$ est maximum, c'est-à-dire si $t = 1$.

II-3 Méthode de quadratisation partielle empirique

Cette méthode est justifiée par deux remarques suivantes :

- 1) Tout d'abord, nous nous plaçons toujours dans le sous-espace $\mathbb{R}^p$ de $\mathbb{R}^n$ tel que l'approchant $[A_p(X^r)]$ du hessien soit strictement positif.

Dans notre étude, nous n'utiliserons que les approchants $[A_p(X^r)]$ et $[B_p(X^r)]$ du hessien et du gradient. Alors, la relation (I, 9) devient :

$$f([X]) \simeq f([X^r]) + [B_p(X^r)]^T [X - X^r] + \frac{1}{2} [X - X^r]^T [A_p(X^r)] [X - X^r] + E([\ell], [X - X^r])$$

où $E([\ell], [X - X^r])$ est le terme d'erreur qui est fonction de la matrice des pas et de $[X - X^r]$.

Les mêmes calculs nous donnent :

$$f([X]) - f([X^r]) = - (t - \frac{t^2}{2}) [X^* - X^r]^T [A_p(X^r)] [X^* - X^r] + E([\ell], [X - X^r]) \quad (I, 14)$$

$$\text{où } [X^*] \text{ vérifie : } [B_p(X^r)] + [A_p(X^r)] [X^* - X^r] = 0$$

Le comportement du terme d'erreur $E([\ell], [X - X^r])$ est mal connu. La seule propriété que nous savons est qu'il devient négligeable devant le premier terme quand $[\ell]$ et t deviennent simultanément petits. Celle-ci nous conduit à la technique empirique suivante :

1a) * Nous calculons $[X^*]$ par $[B_p(X^r)] + [A_p(X^r)] [X^* - X^r] = 0$ pour une $[\ell]$ fixée.

1b) * Puis pour trouver $[X^{r+1}]$ de l'étape suivante, nous essayons successivement avec $t = 1, \frac{1}{2}, \frac{1}{4}, \dots$ en nous contentant de prendre

pour t la première de ces valeurs qui vérifie $f([X^{r+1}]) < f([X^r])$.
Si t devient suffisamment petit et il n'y a toujours pas d'amélioration, nous divisons $[\ell]$ par 2 et recommençons la phase a) après avoir calculé $[A_p(X^r)]$ et $[B_p(X^r)]$ pour cette valeur de $[\ell]$.

2) Considérons maintenant que nous sommes dans l'espace $\mathbb{R}^n$. L'approchant $[A_n(X^r)]$ du hessien n'est pas en général strictement défini positif. La considération de tous les termes du second degré n'est donc plus intéressante. Nous allons chercher un sous-espace $\mathbb{R}^p$ formé de p des n variables pour lequel le hessien $[A_p(X^r)]$ obtenue de $[A_n(X^r)]$ en ne conservant que les lignes et les colonnes correspondant à ces variables est strictement défini positif.

Donc, pour les variables du sous-espace $\mathbb{R}^p$, l'étude précédente est valable et nous prenons comme direction partielle de descente la direction $(x_i^* - x_i^r)$ où $x_i \in \mathbb{R}^p$. Pour les autres variables, nous pouvons concevoir les variantes suivantes :

2a) * Prenons pour valeurs des x_j où $x_j \notin \mathbb{R}^p$ les valeurs de l'étape précédente.

2b) * Ou bien appliquons la méthode du gradient à ces directions (l'approchant $[B_p(X^r)]$ du gradient est utilisé).

Nous notons que la différence entre cette méthode de quadratisation partielle empirique et la méthode précédente réside dans la recherche de $\mathbb{R}^p$, autrement dit il faut extraire de $[A_n(X^r)]$, une $[A_p(X^r)]$ strictement définie positive. Une technique sera proposée en II - 5. La deuxième variante peut se révéler nécessaire dans certains cas. En effet, supposons que nous sommes à la même étape. Il se peut que pour une direction par exemple x_1 , si nous appliquons la première variante, la valeur de x_1 reste inchangée en permanence, donc nous n'arrivons jamais au point optimum.

II - 4 Plan de recherche

Le plan de recherche de cette méthode empirique comprend les principales phases suivantes :

- phase 1 : Initialisation
- phase 2 : Calcul des matrices $[A_n(X^r)]$ et $[B_n(X^r)]$ qui approchent le hessien et le gradient.
- phase 3 : Recherche du sous-espace $\mathbb{R}^p$ en extrayant de $[A_n(X^r)]$ une $[A_p(X^r)]$ strictement définie positive.
- phase 4 : Calculer les nouvelles valeurs de x_i^* où i = indice des variables $\in \mathbb{R}^p$.
- phase 5 : Calcul du nouveau point $[X^{r+1}]$
- phase 6° : Test d'arrêt.

Par rapport à la méthode précédente, nous notons qu'il y a qu'une seule différence au niveau de la troisième phase. Pour cette phase, nous proposons la technique ci-dessous (II-5).

II - 5 Technique proposée pour la recherche de $\mathbb{R}^p$

Cette technique utilise la décomposition d'une matrice symétrique par la méthode de Choleski.

Soient $[A_n]$ une matrice symétrique quelconque :

$$[A_n] = (a_{ij}) \quad \begin{cases} i = 1, 2, \dots, n \\ j = 1, 2, \dots, n \end{cases} \quad a_{ij} = a_{ji}$$

et

$$[R] = \begin{bmatrix} r_{11} & & & \\ r_{12} & r_{22} & & 0 \\ \vdots & & \ddots & \\ r_{1n} & & & r_{nn} \end{bmatrix} \quad r_{ij} = 0 \text{ pour } j > i$$

Considérons $[A_n] = [R] [R]^T$

Cette relation matricielle permet de calculer r_{ij} en fonction des a_{ij} .

En effet nous avons :

$$a_{11} = r_{11}^2 \quad (I, 15)$$

Donc si $a_{11} \leq 0$, la relation (15) ne peut pas être vérifiée, dans ce cas, nous supprimons dans chacune des matrices $[R]$, $[R]^T$, $[A_n]$ la première ligne et la première colonne.

$$\text{Par contre si } a_{11} > 0 \Rightarrow r_{11} = \sqrt{a_{11}}$$

Les autres éléments de la première colonne de $[R]$ sont calculés par les relations :

$$a_{1j} = r_{11} r_{1j} \Rightarrow r_{1j} = \frac{a_{1j}}{r_{11}} = \frac{a_{1j}}{\sqrt{a_{11}}} \quad (I,16)$$

$$j = 2, 3, \dots, n$$

Supposons que nous ayons ainsi traité les $(j-1)$ premières colonnes de $[R]$, c'est-à-dire que ou bien nous en avons calculé les éléments ou bien nous avons rayé la ligne et la colonne, le traitement de la j ème colonne sera comme suit :

$$(I,17) \quad a_{jj} = \sum_{\substack{\ell \in K \\ \ell \leq j}} r_{\ell j}^2 \quad \text{où } K = \text{ensemble des indices des lignes} \\ \text{ou des colonnes non rayées}$$

$$(I,17) \text{ donne : } r_{jj}^2 = a_{jj} - \sum_{\substack{\ell \in K \\ \ell < j}} r_{\ell j}^2 \quad (I,18)$$

Tous les éléments du second membre de (I,18) sont déjà calculés ou connus. Donc si la quantité $a_{jj} - \sum_{\ell \in K, \ell < j} r_{\ell j}^2$ est négative ou nulle, nous supprimons la ligne et la colonne d'indice j de $[R]$ et $[A_n]$ et nous passons à la colonne suivante. Sinon, nous prenons alors :

$$r_{jj} = \sqrt{a_{jj} - \sum_{\substack{\ell \in K \\ \ell < j}} r_{\ell j}^2}$$

et puis nous calculons les autres éléments de la j ème colonne par exemple :

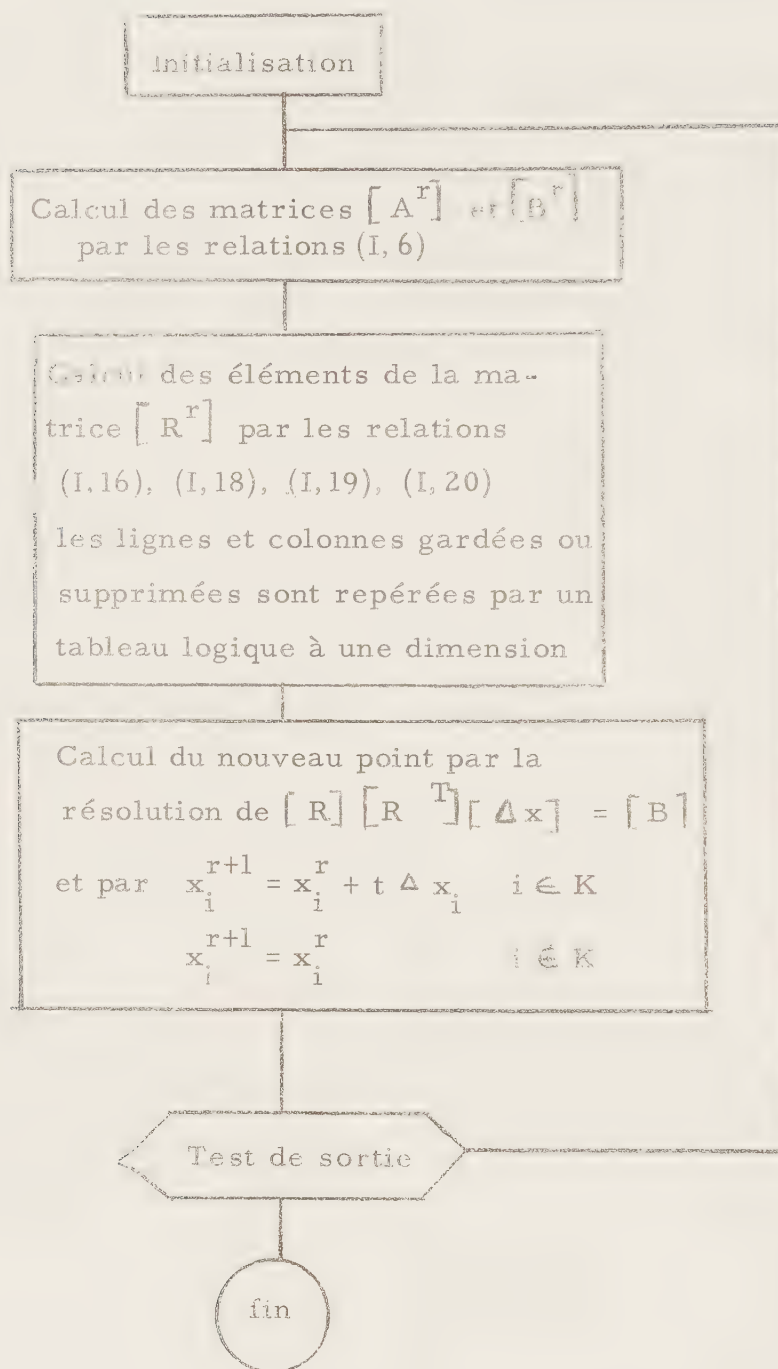
$$k_j = \sum_{\substack{\ell \in K \\ \ell \leq j}} r_{\ell k} r_{\ell j} \quad j \in K, k = j+1, \dots, n \quad (I,20)$$

Dans ces relations, les γ_{lj} sont connus, sauf γ_{jk} pour $k = j+1, \dots, n$, d'où γ_{jk} .

Alors le nombre d'éléments de K est augmenté de 1 et nous recommençons la même technique pour la colonne suivante jusqu'à $j = n$, nous obtenons donc la matrice $[R]$. Le calcul de $[X^{*r} - X^r]$ devient facile par la résolution de deux systèmes triangulaires. Remarquons que si $[A_n]$ est définie positive, toutes les coordonnées seront modifiées comme dans la première méthode.

II - 6 Organigramme

Organigramme



II - 7 Résultats numériques des essais

Les essais sont faits sur trois fonctions définies en (I). Dans les tableaux de résultats, la colonne supplémentaire placée à droite de celle des x_i nous indique si x_i appartient au sous-espace $\mathbb{R}^p$ ou non.

$$\begin{cases} 1 & \Rightarrow x_i & \mathbb{R}^p \\ 0 & \Rightarrow x_i & \mathbb{R}^p \end{cases}$$

II.7.1 Essais sur :

$$f(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2 + e^{x_1^2 + x_2^2} - 1$$

1°) point initial (0.5 ; 0.5) - Pas initial (0.5 ; 0.5)

NR	x_1		x_2		$f([X])$
0	0.5	1	0.5	0	0.1399×10^1
1	0.313×10^{-1}	1	0.5	1	0.5519
2	-0.175	1	0.232×10^{-1}	1	0.6603×10^{-1}
3	-0.146×10^{-1}	1	0.315×10^{-1}	1	0.1947×10^{-2}
4	0.510×10^{-2}	1	-0.492×10^{-2}	1	0.7539×10^{-4}
5	-0.105×10^{-2}	1	0.105×10^{-2}	1	0.3314×10^{-5}
6	0.220×10^{-3}	1	-0.220×10^{-3}	1	0.1457×10^{-6}
7	-0.462×10^{-4}	1	0.462×10^{-4}	1	0.6408×10^{-8}
8	0.203×10^{-5}	1	-0.203×10^{-5}	1	0.1238×10^{-10}
9	-0.214×10^{-7}	1	0.214×10^{-7}	1	0.1348×10^{-14}
10	0.560×10^{-10}		-0.560×10^{-10}		0.3136×10^{-20}

La symétrie de x_1 et x_2 est rompue dès la première étape. En comparant avec le premier essai de I.2.3.2, il semble qu'il vaut mieux ne pas rompre la symétrie dans le cas de cette fonction.

2°) Point initial (1 ; 1)

Pas initial (0.25 ; 0.25)

NR	x_1		x_2		$f(x)$
0	1	1	1	0	0.3187 × 10 ¹
1	0.7500	1	1	0	0.4906 × 10 ¹
2	0.5000	1	1	1	0.2965
3	0.2500	1	0.5000	1	0.9133
4	0.1000	1	0.157	1	0.9103 × 10 ⁻¹
5	0.0000 × 10 ⁻²	1	0.663 × 10 ⁻²	1	0.3190 × 10 ⁻¹
6	-0.440 × 10 ⁻⁴	1	0.769 × 10 ⁻⁴	1	0.1232 × 10 ⁻⁴
7	0.266 × 10 ⁻³	1	-0.265 × 10 ⁻⁵	1	0.2119 × 10 ⁻¹⁰
8	-0.281 × 10 ⁻³	1	0.281 × 10 ⁻⁷	1	0.2341 × 10 ⁻¹⁸
9	0.733 × 10 ⁻¹⁰		0.733 × 10 ⁻¹⁰		0.5370 × 10 ⁻¹⁰

II.7.2 Essais sur :

$$f(x_1, x_2, x_3, x_4) = x_1^2 + 10 x_2^2 + x_3^2 + 5 x_4^2$$

1°) Point initial (2 ; 1.5 ; 1.2)

Pas initial (1 ; 0.5 ; 0.5 ; 1)

NR	x_1		x_2		x_3		x_4		$f(x)$
0	2	1	1.5	1	1	0	1	1	0.1098
1	-0.22 × 10 ⁻¹	1	0.89 × 10 ⁻¹	1	1	1	0.00	1	0.1173
2	0.00	1	0.59 × 10 ⁻¹	1	0.11 × 10 ⁻¹	1	0.00	1	0.1480
3	0.81 × 10 ⁻²	1	-0.40 × 10 ⁻¹	1	0.50 × 10 ⁻²	1	0.12 × 10 ⁻¹	1	0.2409
4	0.83 × 10 ⁻²	1	0.38 × 10 ⁻⁴	1	-0.43 × 10 ⁻⁴	1	0.10 × 10 ⁻¹	1	0.1472
5	0.83 × 10 ⁻¹²	1	-0.49 × 10 ⁻⁶	1	0.48 × 10 ⁻⁵	1	-0.10 × 10 ⁻¹²	1	0.9073
6	0.83 × 10 ⁻¹⁷	1	0.51 × 10 ⁻⁸	1	-0.10 × 10 ⁻⁷	1	-0.10 × 10 ⁻¹⁷	1	0.1432
7	0.93 × 10 ⁻¹⁷		0.52 × 10 ⁻¹¹		-0.23 × 10 ⁻¹⁰		-0.10 × 10 ⁻¹⁷	1	0.7804

2°) Point initial (1 ; 1 ; 1 ; 1)

Pas initial (1 ; 1 ; 1 ; 1)

	x_1		x_2		x_3		x_4		$f([X])$
0	1	1	1	1	1	0	1	1	0.1872×10^2
1	0.00	1	-0.11×10^{-1}	1	1	1	0.00	1	0.9906
2	0.54×10^{-16}	1	0.16	1	0.12	1	0.11×10^{-16}	1	0.2797
3	0.58×10^{-16}	1	-0.45×10^{-2}	1	0.11	1	0.15×10^{-16}	1	0.1145×10^{-1}
4	0.58×10^{-16}	1	-0.56×10^{-2}	1	-0.74×10^{-2}	1	0.18×10^{-16}	1	0.3255×10^{-3}
5	0.58×10^{-16}	1	-0.47×10^{-3}	1	0.24×10^{-2}	1	0.18×10^{-16}	1	0.6841×10^{-5}
6	0.58×10^{-16}	1	0.11×10^{-3}	1	-0.26×10^{-3}	1	0.18×10^{-16}	1	0.1601×10^{-6}
7	0.58×10^{-16}	1	-0.14×10^{-4}	1	0.51×10^{-4}	1	0.18×10^{-16}	1	0.3826×10^{-8}
8	0.26×10^{-17}	1	0.42×10^{-6}	1	-0.12×10^{-5}	1	0.45×10^{-17}	1	0.2622×10^{-11}
9	0.26×10^{-17}	1	0.23×10^{-8}	1	0.78×10^{-8}	1	0.45×10^{-17}	1	0.1130×10^{-15}
10	0.26×10^{-17}		0.36×10^{-11}		-0.11×10^{-10}		0.45×10^{-17}		0.2463×10^{-21}

II.7.3 Essais sur la fonction de 6 variables définie en (I) (page 17)

Nous avons effectué les essais suivants :

1°) Point initial (0.1 ; 1.5 ; 0.0 ; 1.5 ; -1.0 ; 0.5)

Pas initial (0.5 ; 0.5 ; 0.5 ; 0.5 ; 0.5 ; 0.5)

2°) Point initial (0.5 ; 1.0 ; 0.0 ; 1.0 ; -1.3 ; 1.0)

Pas initial (0.25 ; 0.25 ; 0.25 ; 0.25 ; 0.25 ; 0.25)

3°) Point initial (0.5 ; 1.0 ; 0.0 ; 1.0 ; -1.3 ; 1.0)

Pas initial (1.0 ; 1.0 ; 1.0 ; 1.0 ; 1.0 ; 1.0)

Les résultats figurent respectivement dans les tableaux N° 4, 5, 6.

Dans le tableau n° 4 bis, nous détaillons quelques premières étapes du tableau n° 4.

NR	x_1	x_2	x_3	x_4	x_5	x_6	$f([X])$
0	0.1	1	0.0	1.5	-1.0	0.5	0.3520×10^2
1	-0.1656E+00	1	0.0	1.5	-0.5123E+00	0.5	0.2489×10^2
2	-0.5771E-01	1	0.0	1.5	-0.1485E+01	0.5	0.2195×10^2
3	-0.2234E+00	1	0.0	0.1799E+01	-0.1485E+01	0.5	0.1854×10^2
5	-0.3614E+00	1	0.0	0.8906E+00	-0.1076E+01	0.5	0.7461×10
10	-0.1435E+00	1	-0.1388E+00	0.6565E+00	-0.9326E+00	0.6894E+00	0.2238
15	-0.6164E-01	1	-0.3458E-01	0.7446E+00	-0.9326E+00	0.6894E+00	0.1322×10^{-1}
20	-0.4457E-01	1	-0.1218E+00	0.7854E+00	-0.9001E+00	0.6918E+00	0.3741×10^{-2}
25	-0.2992E-01	1	-0.9535E-01	0.7741E+00	-0.9168E+00	0.7263E+00	0.3283×10^{-2}
30	-0.2734E-01	1	-0.1055E+00	0.8231E+00	-0.9840E+00	0.7596E+00	0.3065×10^{-2}
35	-0.2563E-01	1	-0.1318E+00	0.9115E+00	-0.1091E+01	0.8055E+00	0.2784×10^{-2}
40	-0.2157E-01	1	-0.1691E+00	0.1041E+01	-0.1248E+01	0.8755E+00	0.2482×10^{-2}
45	-0.2094E-01	1	-0.1764E+00	0.1066E+01	-0.1277E+01	0.8886E+00	0.2441×10^{-2}
50	-0.1611E-01	1	-0.2289E+00	0.1246E+01	-0.1497E+01	0.9854E+00	0.2288×10^{-2}
53	-0.1573E-01	1	-0.2330E+00	0.1260E+01	-0.1514E+01	0.9930E+00	0.2288×10^{-2}

Tab. 6.30 n° 4

pas = 0.5

Tableau n° 4 bis

NR	NTE*	NPAS	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	f ([X])
0	1	1	0.1000	1.5000	0.0000	1.5000	-1.0000	0.5000	0.3520 10 ²
1	1	1	-0.4313	1.5000	0.0000	1.5000	-0.0246	0.5000	0.4518 10 ²
	2	1	-0.1656	1.5000	0.0000	1.5000	-0.5123	0.5000	0.2489 10 ²
2	1	1	-0.0577	1.5000	0.0000	1.5000	0.1485E+01	0.5000	0.2195 10 ²
3	1	1	-0.7205	1.5000	0.0000	0.2695E+01	0.1485E+01	0.5000	0.5665 10 ²
	2	1	-0.3891	1.5000	0.0000	+0.2097E+01	0.1485E+01	0.5000	0.2374 10 ²
	3	1	-0.2234	1.5000	0.0000	0.1799E+01	0.1485E+01	0.5000	0.1854 10 ²
4	1	1	-0.3130	1.5000	0.0000	0.1463E+01	0.1485E+01	0.5000	0.1254 10 ²
5	1	1	-0.3614	1.5000	0.0000	0.8906	0.1485E+01	0.5000	0.7461 10

* NTE indique le nombre de fois de division de t₀ par 2 pendant l'étape NR correspondant.

Remarque : La comparaison avec le tableau N° 1 bis met en évidence l'amélioration dans l'évolution de f ([X]), en particulier la suppression de la valeur très grande en 5ème ligne.

Tableau n° 5 pas = 0.25

NR	x ₁		x ₂		x ₃		x ₄		x ₅		x ₆		f ([X])
0	0.5	1	1	0	0	1	1	0	-1.3	0	1	1	0.2967 ×10 ²
1	-0.8307E-01	1	1	0	0.1221E+01	1	1	0	-1.3	0	0.5098E+00	0	0.2785 ×10 ²
2	0.1855E+00	1	1	0	0.5966E+00	1	1	0	-1.3	0	0.5098 "	1	0.2300 ×10 ²
3	0.4821 "	1	1	0	-0.1845E+00	1	1	0	-1.3	0	0.1048E+01	1	0.2266 ×10 ²
4	0.5222E-01	1	1	0	0.6938 "	1	1	0	-1.3	0	0.8057E+00	0	0.1087 ×10 ²
5	0.1850E+00	1	1	0	0.1678 "	1	1	0	-1.3	0	0.8057 "	1	0.7040 ×10
8	-0.6222E-01	1	1	0	0.1622 "	1	1	0	-0.1473E+01	1	0.9228 "	1	0.6925
12	0.8476 "	1	0.9201E+00	0	0.1563E-01	1	1	0	-0.1184 "	0	0.9767 "	1	0.1158
16	0.3496 "	1	0.1001E+01	1	-0.1004E+00	1	0.1063E+01	0	-0.1248 "	0	0.9503 "	1	0.3399 ×10 ⁻¹
20	0.2374 "	1	0.1007 "	1	-0.1157 "	1	0.1063 "	1	-0.1272 "	0	0.9521 "	1	0.1968
24	0.3955E-02	1	0.9984E+00	1	-0.1292 "	1	0.1063 "	0	-0.1292 "	1	0.9304 "	1	0.6733 ×10 ⁻²
28	-0.2212E-01	1	0.1006E+01	1	-0.1557 "	1	0.1055 "	1	-0.1298 "	1	0.9033 "	1	0.2606 ×10 ⁻²
32	-0.2054 "	1	0.1007 "	1	-0.1791 "	1	0.1079 "	1	-0.1297 "	1	0.8980 "	1	0.2418 ×10 ⁻²
36	-0.1953 "	1	0.1009 "	1	-0.1919 "	1	0.1119 "	1	-0.1342 "	1	0.9171 "	1	0.2368 ×10 ⁻²
40	-0.1604 "	1	0.1012 "	1	-0.2297 "	1	0.1249 "	1	-0.1497 "	1	0.9868 "	1	0.2288 ×10 ⁻²
44	-0.1673E-01		0.1012E+01		-0.2310 "		0.1260E+01		-0.1514E+01		0.9939		0.2288 ×10 ⁻²

Tableau n° 6

pas = 1

NR	x ₁		x ₂		x ₃		x ₄		x ₅		x ₆		f(x)
0	0.5	1	1	0	0	0	1	0	-1.3	0	1	0	0.2967 E+02
1	0.1694E+00	1	1	0	0	0	1	0	-0.1026E+00	0	1	0	0.8300 E+01
2	0.3193 "	1	1	0	0	0	1	0	-0.9129 "	1	1	1	0.6560 "
3	0.1460 "	1	1	0	0	0	1	0	-0.9129 "	0	0.1460E+01	0	0.2995 "
4	0.1976 "	1	1	0	0	0	1	0	-0.1353E+01	1	0.1460 "	1	0.3880 E+00
5	0.1849 "	1	1	1	0	0	1	0	-0.1353 "	1	0.1483 "	1	0.3425 "
8	0.1247 "	1	0.1129E+01	1	-0.9109E-01	0	1	0	-0.1549 "	1	0.1542 "	1	0.1481 "
12	0.8578 "	1	0.1110E+01	1	-0.9109 "	0	0.1020E+01	0	-0.1553E+01	1	0.1430 "	1	0.1019 "
16	0.3657 "	1	0.9908E+00	1	-0.9109 "	1	0.1242 "	0	-0.1669 "	1	0.1196 "	1	0.1250 E-01
20	0.3007E-01	1	0.1004E+01	1	-0.1282E+00	1	0.1277 "	1	-0.1695 "	0	0.1195 "	0	0.8670 E-02
24	-0.1825E-04	1	0.9868E+00	1	-0.1049 "	1	0.1253 "	0	-0.1722 "	1	0.1191 "	1	0.6044 "
28	-0.7225E-04	1	0.9980 "	1	-0.1634 "	1	0.1309 "	1	-0.1735 "	1	0.1153 "	1	0.4423 "
32	-0.8399E-02	1	0.1009E+01	1	-0.2344 "	1	0.1382 "	1	-0.1733 "	1	0.1114 "	1	0.2839 "
36	-0.1054E-01	1	0.1016 "	1	-0.2717 "	1	0.1421 "	1	-0.1726 "	1	0.1094 "	1	0.2439 "
40	-0.1143 "	1	0.1017 "	1	-0.2770 "	1	0.1414 "	1	-0.1701 "	1	0.1076 "	1	0.2382 "
44	-0.1480 "	1	0.1013 "	1	-0.2427 "	1	0.1294 "	1	-0.1555 "	1	0.1011 "	1	0.2292 "
49	-0.1573E-01		0.1012E+01		-0.2330E+00		0.1260E+01		-0.1514E+01		0.9930 "		0.2288 E-02

II.8 Méthode mixte de quadratisation partielle et du gradient

Soit la fonction $f(X)$, $X = (x_1, x_2, \dots, x_i, \dots, x_n) \in \mathbb{R}^n$

Les notations $[H_n(X^r)]$, $[G_n(X^r)]$, $[H_p(X^r)]$ et $[G_p(X^r)]$

ont la même signification que précédemment.

Notons $[G_{n-p}(X^r)]$ = vecteur gradient de f par rapport aux variables

$$x_j \in \mathbb{R}^p.$$

$[X_p]$ = vecteur colonne dont les composantes $= x_i \in \mathbb{R}^p$

$[X_{n-p}]$ = " " " " $= x_j \in \mathbb{R}^p$.

II.8.1 Proposition

Si $[X_p^*]$ est solution de $[H_p(X^r)] [X_p - X_p^r] + [G_p(X^r)] = 0$
la direction :
$$[X^* - X^r] = \begin{bmatrix} [X_p^* - X_p^r] \\ [G_{n-p}(X^r)] \end{bmatrix}$$
 est une direction de descente.

Considérons 2 segments définis par :

$$[X_p] = [X_p^r] + t [X_p^* - X_p^r]$$

$$[X_{n-p}] = [X_{n-p}^r] + \lambda [G_{n-p}(X_{n-p}^r)]$$

Développons $f([X]) = f([X_p], [X_{n-p}])$ jusqu'au second ordre par rapport à $[X_p]$ au voisinage de $[X_p^r]$.

$$\begin{aligned} f([X_p], [X_{n-p}]) &= f([X_p^r] + t [X_p^* - X_p^r], [X_{n-p}]) \\ &= f([X_p^r], [X_{n-p}]) + [G_p(X^r)]^T [X_p - X_p^r] + \\ &\quad \frac{1}{2} [X_p - X_p^r]^T [H_p(X^r)] [X_p - X_p^r] + 0 \parallel [X_p - X_p^r] \parallel \end{aligned}$$

Nous avons vu en II.2 que si t devient suffisamment petit, alors

$$f([X_p], [X_{n-p}]) < f([X_p^r], [X_{n-p}]) \quad (I, 21)$$

Maintenant, appliquons la méthode du gradient aux variables $[X_{n-p}]$, c'est-à-dire développons $f([X_p^r], [X_{n-p}])$ jusqu'au premier ordre :

$$\begin{aligned}
 f([X_p^r], [X_{n-p}]) &= f([X_p^r, X_{n-p}^r + \lambda [G_{n-p}(X^r)]] \\
 &= f([X_p^r], [X_{n-p}^r]) + [G_{n-p}(X^r)]^T [X_{n-p} - X_{n-p}^r] \\
 &\quad + 0 \|X_{n-p} - X_{n-p}^r\|^2 \\
 &= f([X_p^r], [X_{n-p}^r]) + \lambda [G_{n-p}(X^r)]^T [G_{n-p}(X^r)] \\
 &\quad + 0 \|X_{n-p} - X_{n-p}^r\|^2
 \end{aligned}$$

donc pour $\lambda < 0$ et de valeur absolue suffisamment petite,

$$f([X_p^r], [X_{n-p}]) < f([X_p^r], [X_{n-p}^r]) \quad (I, 22)$$

Les inégalités (I, 21), (I, 22) nous prouvent qu'à chaque point $[X^r]$, il existe au moins un couple (t', λ') tel que

$$f([X_p^r + t' (X_p^* - X_p^r)], [X_{n-p}^r + \lambda' [G_{n-p}(X^r)]]) < f([X_p^r], [X_{n-p}^r]) \quad (I, 23)$$

Comme dans la méthode de quadratisation partielle, dans le cas où nous utilisons les approchants du hessien et du gradient, le terme d'erreur de la méthode mixte devient encore plus compliqué car il est encore fonction de $\lambda [X_{n-p} - X_{n-p}^r]$, par contre sa propriété connue reste toujours valable, à savoir que cette erreur est négligeable devant le premier terme :

$$-(t - \frac{t^2}{2}) [X_p^* - X_p^r]^T [H_p(X^r)] [X_p^* - X_p^r] + \lambda [G_{n-p}(X^r)]^T [G_{n-p}(X^r)]$$

quand la matrice des $[\ell]$, t et λ deviennent simultanément petits. Elle nous amène à procéder de la même façon en divisant le couple (t, λ) par 2 au lieu de diviser uniquement t .

II.8.2. Résultats numériques

Nous faisons des essais sur la fonction de 6 variables définie en (I) afin de comparer avec la méthode exposée en (I) et la méthode de quadratisation partielle.

Dans le tableau des résultats, à droite des colonnes des x_i , il y a une petite colonne qui nous indique l'espace auquel appartient la variable x_i .

1 $\iff$ variable x_i modifiée par la technique de quadratisation

0 $\iff$ variable x_i modifiée par la technique du gradient.

Tableau n° 7 $\text{pas} = 0.5, \lambda_0 = 5 \times 10^{-3}$

NR	x_1	x_2	x_3	x_4	x_5	x_6	$f(X)$				
0	0.1	1	1.5	0	0.0	0	-1.0	1	0.5	0	0.3520×10^2
1	-0.1656E+00	1	0.1218E+01	0	-0.1738E+00	0	0.1409E+01	0	-0.5123E+00	1	$0.5524E+00$
2	0.9218E-02	1	0.1191E+01	0	-0.2153E+00	1	0.1366E+01	0	-0.1586E+01	0	0.1099E+01
3	0.1583E+00	1	0.1162E+01	0	-0.5577E+00	0	0.1360E+01	1	-0.1583E+01	0	0.1353E+01
4	0.8141E-01	1	0.1159E+01	0	-0.5565E+00	1	0.1582E+01	0	-0.1580E+01	0	0.1226E+01
5	0.4424E-01	1	0.1159E+01	1	-0.5046E+00	0	0.1582E+01	0	-0.1581E+01	1	0.1160E+01
10	0.4214E-01	1	0.1078E+01	1	-0.4744E+00	0	0.1660E+01	0	-0.1597E+01	1	0.1067E+01
20	-0.6654E-01	1	0.1028E+01	1	-0.3865E+00	1	0.1524E+01	1	-0.1738E+01	0	$0.9904E+00$
25	-0.2353E-01	1	0.1025E+01	1	-0.3503E+00	1	0.1505E+01	1	-0.1719E+01	1	0.1038E+01
30	-0.1609E-01	1	0.1020E+01	1	-0.3039E+00	1	0.1452E+01	1	-0.1717E+01	1	0.1067E+01
35	-0.1170E-01	1	0.1016E+01	1	-0.2754E+00	1	0.1407E+01	1	-0.1692E+01	1	0.1072E+01
40	-0.1544E-01	1	0.1013E+01	1	-0.2360E+00	1	0.1271E+01	1	-0.1526E+01	1	0.9985E+00
43	-0.1572E-01		0.1012E+01		-0.2330E+00		0.1260E+01		-0.1514E+01		0.9930E+00

Tableau n° 7 bis

NR	NTE	NPAS	x_1	x_2	x_3	x_4	x_5	x_6	f (X)
0	1	1	0.1000	1	0	1.5000	0	-1.0000	0.3520 ×10 ²
	1	1	-0.4313	0.9351	-0.3477	0.1318E+01	-0.0246	0.5647	0.5070 ×10 ²
1	2	1	-0.1656	1	0	0.1218E+01	0	-0.5123	0.9981 ×10 ²
	1	1	0.1841	0.1165E+01	-0.2568	0.1322E+01	-0.2660E+01	0.1665E+01	0.1017 ×10 ²
2	2	1	0.0092	1	0	0.1191E+01	0	-0.1586E+01	0.8730
	1	1	-0.3810	0.1113E+01	-0.2522	0.2160E+01	-0.1575E+01	0.5811	0.1123 ×10 ²
3	2	1	-0.1859	0.1152E+01	-0.2337	0.1763E+01	-0.1581E+01	0.8400	0.3828 ×10
	3	1	-0.0883	0.1172E+01	-0.2245	0.1564E+01	-0.1584E+01	0.9695	0.1682 ×10
4	4	1	-0.0396	0.1181E+01	-0.2199	0.1465E+01	-0.1585E+01	0.1034E+01	0.1094 ×10
	5	1	-0.0152	1	0	0.1186E+01	-0.1586E+01	0.1067E+01	0.9353
1	1	2	0.3074	0.1133E+01	-0.9001	0.1355E+01	-0.1580E+01	0.1607E+01	0.5434 ×10
	2	2	0.1583	0.1162E+01	-0.5577	0.1360E+01	-0.1583E+01	0.1353E+01	0.7741
4	1	1	0.4570	1	0	0.1139E+01	-0.15479	0	0.2776 ×10 ²
	2	1	-0.1493	0.1151E+01	-0.5528	0.2246E+01	-0.1572E+01	0.8454	0.5768 ×10
3	1	1	0.0045	0.1156E+01	-0.5552	0.1803E+01	-0.1576E+01	0.1099E+01	0.8381
	4	1	0.0814	0.1159E+01	-0.5564	0.1582E+01	-0.1580E+01	0.1226E+01	0.1504

Tableau n° 8 pas = 0.25 , $\lambda_0 = 2.5 \times 10^{-5}$

NR	x_1		x_2		x_3		x_4		x_5		x_6		f (X)
0	0.5	1	1.0	0	0.0	0	1.0	0	-1.3	1	1	0	0.2967×10^2
1	0.1694E+00	1	0.9277E+00	0	-0.2903E-01	0	0.9967E+00	0	-0.1028E+01	1	0.1028E+01	0	0.7446×10
2	0.2650E+00	1	0.8598E+00	0	-0.1399E+00	0	0.8779E+00	1	-0.5986E+00	0	0.9406E+00	0	0.1447×10
3	0.2356E+00	1	0.8567E+00	0	-0.1408E+00	1	0.9808E+00	0	-0.5977E+00	0	0.9429E+00	0	0.1344×10
4	0.1746E+00	1	0.9206E+00	0	0.1869E-01	1	0.9620E+00	0	-0.6290E+00	0	0.9091E+00	0	0.1191×19
5	0.2042E+00	1	0.8943E+00	1	-0.8554E-01	0	0.9264E+00	0	-0.6556E+00	1	0.8948E+00	1	0.7747
8	0.1132E+00	1	0.1056E+01	0	-0.2893E+00	0	0.9052E+00	1	-0.9979E+00	1	0.1019E+01	1	0.3223
11	-0.05162E-01	1	0.1068E+01	1	-0.2994E+00	0	0.1214E+01	1	-0.1337E+01	0	0.8720E+00	0	0.3341×10^{-1}
15	-0.04197E-01	1	0.1018E+01	1	-0.2710E+00	1	0.1217E+01	1	-0.1371E+01	1	0.8714E+00	1	0.3387×10^{-2}
18	-0.02847E-01	1	0.1015E+01	1	-0.2499E+00	1	0.1197E+01	1	-0.1368E+01	1	0.8944E+00	1	0.2721×10^{-2}
21	-0.02770E-01	1	0.1013E+01	1	-0.2328E+00	1	0.1178E+01	1	-0.1372E+01	1	0.9050E+00	1	0.2513×10^{-2}
24	-0.02193E-01	1	0.1011E+01	1	-0.2128E+00	1	0.1159E+01	1	-0.1374E+01	1	0.9216E+00	1	0.2366×10^{-2}
27	-0.01858E-01	1	0.1010E+01	1	-0.2035E+00	1	0.1158E+01	1	-0.1389E+01	1	0.9375E+00	1	0.2330×10^{-2}
30	-0.01701E-01		0.1011E+01		-0.2194E+00		0.1213E+01		-0.1456E+01		0.9676E+00		0.2297×10^{-2}
34	-0.01573E-01		0.1012E+01		-0.2329E+00		0.1260E+01		-0.1514E+01		0.9929E+00		0.2288×10^{-2}

II - 9 Temps de calcul de certains essais

Les essais sur la fonction de 6 variables définie à la page
donnent le tableau suivant :

Point initial	Pas initial	méthode de quadratisation par interpolation	méthode de quadratisation partielle	méthode de quadratisation mixte
$x_1 = 0.1 ; x_2 = 1.5$ $x_3 = 0.0 ; x_4 = 1.5$ $x_5 = -1.0 ; x_6 = 0.5$	0.5	4,151 min	2,855 min	3,114 min (pour $\lambda_0 = 5 \cdot 10^{-3}$)
$x_1 = 0.5 ; x_2 = 1.0$ $x_3 = 0.0 ; x_4 = 1.0$ $x_5 = -1.3 ; x_6 = 1.0$	0.25	2,367 min	2,324 min	2,546 min (pour $\lambda_0 = 2.5 \cdot 10^{-2}$)

III - CONCLUSION

Les essais effectués dans cette partie nous permettent de formuler quelques remarques suivantes :

1) Pour les trois méthodes, le temps de calcul dépend évidemment de leurs paramètres :

i) Pour la méthode de quadratisation par interpolation et de quadratisation partielle :

- point initial
- pas initial

ii) Pour la méthode mixte :

- point initial
- pas initial
- valeur initiale (t_0, λ_0) du couple (t, λ)

Le choix des valeurs de ces paramètres pose un problème très délicat, surtout celui des pas initiaux. Une valeur trop petite peut provoquer la non-convergence et par contre une valeur trop grande peut entraîner une augmentation considérable du temps de calcul. Il en est de même pour le choix de (t_0, λ_0) de la méthode mixte. Nous notons que pour la valeur initiale des variables et des pas du tableau n° 1 bis, nous avons essayé avec $t_0 = \lambda_0 = 1$, la convergence est très lente, après 5,148 minutes de calcul, nous n'obtenons que le point M_1 où $f(M_1) = 0.98 \times 10^{-1}$.

Dans les applications pratiques, nous pensons que c'est aux utilisateurs de chercher avec des essais, les valeurs initiales des paramètres appropriés à leurs problèmes particuliers.

2) La convergence de la méthode de quadratisation partielle et mixte est en général plus régulière que celle de la méthode de quadratisation par interpolation. (Tableaux n° 1 bis, 4 bis, 7 bis). De plus, leur performance paraît meilleure que celle de la dernière (II-9). En comparant les deux premières, les résultats semblent assez décevants. Cependant, notons toutefois que la méthode mixte peut devenir nécessaire pour "dépanner" la méthode de quadratisation partielle en cas de blocage de certaines variables.

DEUXIEME PARTIE

ETUDE DU CAS DES VARIABLES LIEES

I - INTRODUCTION ET POSITION DU PROBLEME

Soit f une fonction réelle de n variables :

$$f([X]) \quad [X] \in \mathbb{R}^n \quad (\text{II}, 1)$$

Soit (D) un domaine de $\mathbb{R}^n$ défini par les égalités et les inégalités suivantes :

$$\varphi_\ell([X]) = 0 \quad \ell = 1, 2, \dots, p \quad (\text{II}, 2)$$

$$\varphi_j([X]) \geq 0 \quad j = p+1, \dots, q \quad (\text{II}, 3)$$

Nous supposons que ces fonctions sont une fois continûment dérivables sur (D) .

Le problème consiste à trouver un optimum local $[X^*]$ de $f([X])$ tel que $[X^*]$ vérifie les relations (II, 2) et (II, 3).

Sous forme condensée, nous pouvons écrire :

$$\text{Opt. } \left\{ f([X]) / [X] \in (D) \right\}$$

$$\text{où } (D) = \left\{ X / \varphi_\ell([X]) = 0 \quad \ell = 1, \dots, p \text{ et } \varphi_j([X]) \geq 0 \quad j = p+1, q \right\}$$

- la fonction $f([X])$ est appelée "fonction objectif"
- les $\varphi_\ell([X]) = 0 \quad \ell = 1, \dots, p$ sont appelées "liaisons"
- les $\varphi_j([X]) \geq 0 \quad j = p+1, \dots, q$ sont appelées "contraintes"
- un point $[X^r]$ quelconque $\in (D)$ " " "point admissible"
- une contrainte φ_j est dite saturée (ou active) en un point $[X^r]$ si $\varphi_j([X^r]) = 0$.

Dans cette partie, nous exposerons une technique itérative pour résoudre ce problème d'optimisation avec contraintes. Cette technique est une technique d'élimination numérique basée sur la méthode de quadratisation par-tille exposée dans la première partie et sur la méthode du gradient projeté qui sera décrite au paragraphe II. L'idée de base de cette technique est de ramener un problème d'optimisation avec contraintes à un problème sans contrainte par rapport aux variables libres (hors base) auxquelles nous pouvons appliquer l'une des deux méthodes mentionnées ci-dessus. Les conditions de Kühn et Tucker seront utilisées pour la reconnaissance du point optimun local.

Au point de vue de programmation, nous signalons que la technique d'élimination numérique est également utilisée dans le travail cité en [II], beaucoup de sous-programmes écrits dans ce travail sont donc utilisables dans notre technique. Nous les avons utilisés en nous efforçant d'utiliser les mêmes notations.

Remarques :

- 1) Toutes les autres formes de contraintes peuvent se ramener à la forme canonique décrite ci-dessus.

$$\text{Exemples : } * \varphi([X]) \geq C \iff \varphi([X]) - C \geq 0$$

$$* \varphi_1([X]) \geq \varphi_2([X]) \iff \varphi_1([X]) - \varphi_2([X]) \geq 0$$

$$* \varphi([X]) \leq 0 \iff -\varphi([X]) \geq 0$$

- 2) Pour simplifier l'écriture, nous limiterons dans la suite au cas de maximisation. Dans le cas des problèmes de recherche du minimum, nous nous ramèneront au cas du maximum en se servant de la propriété suivante :

$$\text{Min } [f([X]) / [X] \in (D)] = - \text{Max. } [-f([X]) / [X] \in (D)]$$

II - EXPOSE DE LA METHODE DU GRADIENT PROJETE

Dans cette étude, à la fin de chaque étape, nous avons à tester le signe des coefficients de Kuhn et Tucker. En effectuant ce test, si nous nous trouvons dans le cas où il y a au moins un coefficient positif, quelle que soit la règle de libération des contraintes saturées, rien ne permet d'affirmer que la direction obtenue par la méthode de quadratisation partielle sera compatible avec les contraintes qui viennent d'être libérées. Cette raison nous amène à introduire la méthode du gradient projeté, exposée dans ce paragraphe. Nous démontrons qu'en libérant la contrainte correspondant à un seul coefficient positif par exemple celui de plus grande valeur, cette dernière méthode nous donne une direction de montée compatible avec cette contrainte.

En chaque point $[X]$ nous désignerons par :

L = ensemble des indices des contraintes bilaérales (liaisons)

K = ensemble des indices des contraintes unilatérales saturées en ce point $[X]$

H = ensemble des indices des contraintes unilatérales non saturées.

Alors en ce point :

$$\varphi_{\ell}([X]) = 0 \quad \ell \in L \text{ (II, 4)}$$

$$\varphi_k([X]) = 0 \quad k \in K \text{ (II, 5)}$$

$$\varphi_h([X]) > 0 \quad h \in H \text{ (II, 6).}$$

Le problème est de trouver une direction $[D]$ telle que :

$$f([X] + t[D]) > f([X]) \quad \text{(II, 7)}$$

$$\varphi_{\ell}([X] + t[D]) = 0 \quad \ell \in L \quad \text{(II, 8)}$$

$$\varphi_k([X] + t[D]) \geq 0 \quad k \in K \quad \text{(II, 9)}$$

$$\varphi_h([X] + t[D]) \geq 0 \quad h \in H \quad \text{(II, 10)}$$

Pour une valeur de t suffisamment petite, nous pouvons remplacer les fonctions par leur développement de Taylor du premier ordre. En tenant compte des relations (II, 4), (II, 5), (II, 6), les relations (II, 7), (II, 8), (II, 9), (II, 10) deviennent donc :

$$\Omega = [D]^T [G] > 0 \quad \text{(II, 11)}$$

$$[D]^T [\varphi_{\ell}] = 0 \quad \ell \in L \quad \text{(II, 12)}$$

$$[D]^T [\varphi_k] \geq 0 \quad k \in K \quad \text{(II, 13)}$$

$$\varphi_h([X]) + t[D][\varphi_k] \geq 0 \quad h \in H \quad \text{(II, 14)}$$

$$\text{où } \begin{cases} [G] = \overrightarrow{\text{grad } f([X])} \\ [\varphi_i] = \overrightarrow{\text{grad } \varphi_i([X])} \end{cases}$$

II - 1 Coefficients de Kuhn - Tucker

Au point $[X]$, considérons la direction $[D^*]$ définie comme

suit :

$$[D^*] = \overrightarrow{\text{grad } f([X])} - \sum_{i \in L \cup K} \lambda_i \overrightarrow{\text{grad } \varphi_i([X])}$$

où les coefficients λ_i sont tels que la norme euclidienne $\|D^*\|$ de $[D^*]$ soit minimale.

Posons $[\Lambda] = (\lambda_i)$ et :

$$[\Psi] = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x_1} & \dots & \frac{\partial \varphi_i}{\partial x_1} & \dots \\ \frac{\partial \varphi_1}{\partial x_2} & \dots & \frac{\partial \varphi_i}{\partial x_2} & \dots \\ \vdots & & \vdots & \\ \frac{\partial \varphi_1}{\partial x_n} & \dots & \frac{\partial \varphi_i}{\partial x_n} & \dots \end{bmatrix} = \left[[\varphi_1] \dots [\varphi_i] \right]_{i \in L \cup K}$$

nous avons :

$$\begin{aligned} [D^*] &= [G] - [\Psi][\Lambda] \\ \|D^*\| &= \|[G] - [\Psi][\Lambda]\| \end{aligned} \quad (II, 15)$$

La condition

$$\frac{\partial (\|D^*\|)}{\partial \lambda_i} = 0 \quad \text{entraîne :}$$

$$[B][\Lambda] - [E] = 0 \quad (II, 16)$$

ou

$$[B] = (b_{ii}') \text{ avec } b_{ii}' = [\varphi_i]^T [\varphi_i] = \sum_{j=1}^n \frac{\partial \varphi}{\partial x_j} \frac{\partial \varphi_i}{\partial x_j}$$

$[B]$ est symétrique, définie, positive.

$$E = (e_i) \text{ avec } e_i = [G]^T [\varphi_i] = \sum_{j=1}^n \frac{\partial f}{\partial x_j} \cdot \frac{\partial \varphi_i}{\partial x_j}$$

Démontrons que $[D^*]$ ainsi définie vérifie les conditions (II, 12) et (II, 13). En effet :

$$\begin{aligned} [D^*]^T [\Psi] &= \{ [G] - [\Psi][\Lambda] \}^T [\Psi] \\ &= [G]^T [\Psi] - [\Lambda]^T [\Psi]^T [\Psi] \\ &= [E]^T - [\Lambda]^T [B]^T \\ &= \{ [E] - [B][\Lambda] \}^T \end{aligned}$$

$$(II, 16) \implies [D^*][\Psi] = 0 \quad (II, 17)$$

Dans le cas où $[X]$ est un maximum local, les coefficients λ_ℓ ($\ell \in L$) et λ_k ($k \in K$) deviennent respectivement les multiplicateurs de Lagrange et de Kühn et Tucker.

Nous signalons qu'en pratique, pour calculer ces coefficients, au lieu de construire les deux matrices $[B]$ et $[E]$ et de résoudre (II, 16), nous cherchons la pseudo-solution du système surdéterminé de n équations à s inconnues ($s = \text{card}(L) + \text{card}(K)$) afin de réduire les erreurs de chute :

$$[G] - [\Psi][\underline{\lambda}] = 0 \Rightarrow [\Psi][\underline{\lambda}] = [G] \quad (\text{II, 18})$$

par la procédure MCADIR.

Avec cette direction $[D^*]$, la variation de la fonction :

$$\begin{aligned} \Omega^* &= [D^*]^T [G] \\ &= [D^*]^T \{ [D^*] + [\Psi][\underline{\lambda}] \} \\ &= [D^*]^T [D^*] + [D^*]^T [\Psi][\underline{\lambda}] \\ \Omega^* &= \|D^*\|^2 \quad [\text{par (II, 17)}] \end{aligned}$$

D'où $\Omega^* \geq 0$

{ La direction $[D^*] = [G] - [\Psi][\underline{\lambda}]$ est donc une direction de montée.

II - 2 Caractérisation d'un maximum local

II. 2.1 Calcul de Ω pour une direction $[D]$ quelconque

Soit D une direction quelconque compatible avec

$$\begin{aligned} [D]^T [\Psi_\ell] &= 0 & \ell \in L \\ [D]^T [\Psi_k] &\geq 0 & k \in K \end{aligned}$$

Introduisons la matrice des variables d'écart $[Z]$

$$[Z] = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ Z_1 \\ Z_2 \\ \vdots \\ Z_k \\ \vdots \end{bmatrix} \quad \begin{array}{l} \text{les } p \text{ premiers éléments sont nuls} \\ (\text{Rappelons que } p = \text{Card}(L)) \\ k \in K \end{array}$$

où les z_k sont essentiellement positifs ou nuls.

Nous voyons que :

faire $\begin{cases} z_k = 0 \iff \text{conserver la contrainte d'indice } k \text{ saturée} \\ z_k > 0 \iff \text{libérer cette contrainte} \end{cases}$

La direction $[D]$ sera : $[D] = [G] - [\psi] [\rho]$

où $[\rho] = (\rho_i)$ est solution du système linéaire :

$$[B][\rho] = [E] - [Z] \quad (\text{II}, 19)$$

où $[B]$ et $[E]$ ont les mêmes significations que dans II - 1. En éliminant $[E]$ entre (II, 16) et (II, 19), nous obtenons :

$$[B]\{[\lambda] - [\rho]\} = [Z] \quad (\text{II}, 20)$$

Nous savons que par définition :

$$\begin{aligned} \Omega &= [D]^T [G] \\ &= \{[D]^T - [D^*]^T\} [G] + [D^*]^T [G] \end{aligned}$$

Or $\begin{aligned} [D] &= [G] - [\psi] [\rho] \\ [D^*] &= [G] - [\psi] [\lambda] \end{aligned}$

$$\begin{aligned} \Rightarrow \{[D]^T - [D^*]^T\} [G] &= \{[\lambda]^T - [\rho]^T\} [\psi]^T [G] \\ &= \{[\lambda]^T - [\rho]^T\} [E] \\ &= \{([B]^{-1} [Z])^T\} [B] [\lambda] \\ &= [Z]^T [B]^{-1} [B] [\lambda] \text{ car } [B] \text{ est symétrique} \\ &= [Z]^T [\lambda] \end{aligned}$$

D'où

$$\begin{aligned} \Omega &= \Omega^* + [Z]^T [\lambda] \\ \Omega &= \Omega^* + \sum_{k \in K} \lambda_k z_k \end{aligned} \quad (\text{II}, 21)$$

II.2.2 Caractérisation du maximum local, conditions de Kuhn et Tucker

Pour que $[X]$ soit un maximum local, il faut qu'il n'existe aucune direction $[D]$ de montée acceptable en ce point, c'est-à-dire qu'il n'existe aucune direction $[D]$ compatible avec :

$$\begin{aligned} [D]^T [\psi_\ell] &= 0 & \ell \in L \\ [D]^T [\lambda_k] &\geq 0 & k \in K \end{aligned}$$

telle que $\Omega \geq 0$

La relation (39) montre que pour que ces conditions soient vérifiées, il faut que :

$$i) \Omega^* = 0$$

$$ii) \lambda_k \leq 0 \quad \forall k \in K$$

Autrement dit, la condition nécessaire pour que $[X^*]$ soit un maximum est qu'il existe des nombres λ_ℓ et λ_k tels que :

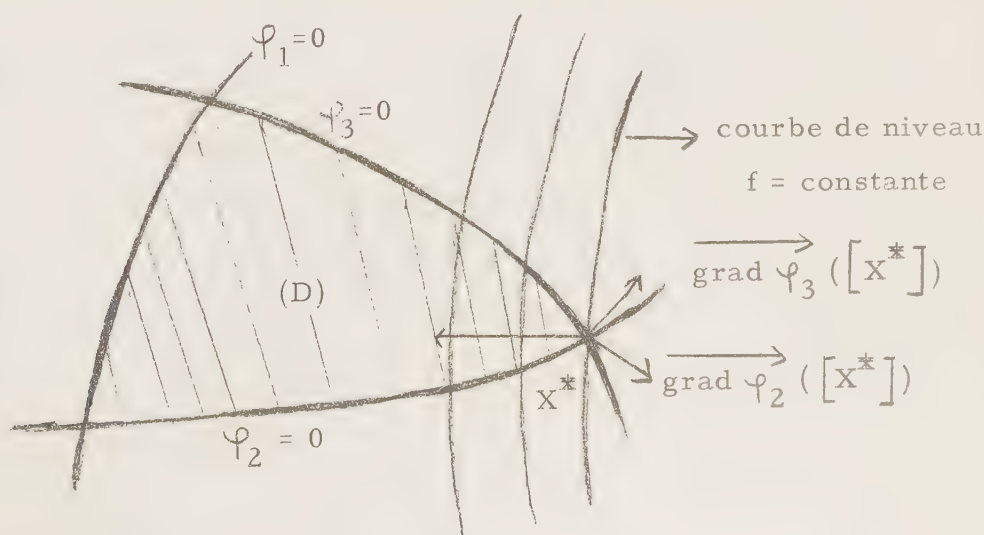
$$\left[\begin{array}{l} 1) \lambda_k \leq 0 \quad \forall k \in K \\ 2) [G^*] = [\psi^*] [\lambda^*] \end{array} \right. \quad (II, 22)$$

$$\Leftrightarrow \frac{\partial f([X^*])}{\partial x_j} = \sum_{\ell \in L} \frac{\partial \psi_\ell([X^*])}{\partial x_j} + \sum_{k \in K} \lambda_k \frac{\partial_k([X^*])}{\partial x_j} \quad j = 1, 2, \dots, n \quad (II, 23)$$

géométriquement, la condition de Kuhn - Tucker se traduit comme suit :

"Au maximum local, le vecteur gradient de la fonction est une combinaison linéaire quelconque des vecteurs gradient des liaisons et non-positive des vecteurs gradient des contraintes saturées".

La condition de non-positivité se traduit par un angle obtenus des 2 vecteurs $\overrightarrow{\text{grad } f}([X^*])$ et $\overrightarrow{\text{grad } \varphi_k}([X^*])$ sur la figure suivante :



Remarques

- 1) La relation (II, 23) représente un système de n équations à $s = \text{Card}(L) + \text{Card}(K)$ inconnues. En général $s < n$, donc il faut tester la compatibilité de ce système dans le test final pour savoir si le point optimum local X^* est atteint. En même temps, il faut tester la condition (II, 22) de la non-positivité des λ_k ($k \in K$).
- 2) Si $K = H = \emptyset$, nous sommes dans le cas sans contraintes unilatérales, la condition (II, 22) n'existe pas.

II - 3 Technique de libération des contraintes saturées

Si au point $[X]$, un ou plusieurs $\lambda_k > 0$ ($k \in K$), nous choisirons l'un d'eux, le plus grand par exemple. Soit k_2 l'indice de ce coefficient

Posons $K' = K - \{k_2\}$

Nous garderons comme contraintes saturées pour le calcul du point suivant les contraintes dont les indices appartiennent à K' , donc la seule contrainte d'indice k_2 est libérée. Nous prenons la direction de montée $[D']$ définie par :

$$[D'] = \overrightarrow{\text{grad } f}([X]) - \sum_{i' \in L \cup K'} \rho_{i'} \overrightarrow{\text{grad } \varphi_{i'}}([X]) \quad (\text{II, 24})$$

Nous allons démontrer que le fait de libérer la seule contrainte pour laquelle $\lambda_{k_2} > 0$ entraîne :

$$[D']^T [\psi_{k_2}] = z_{k_2} > 0$$

donc que la direction $[D']$ définie par (II, 24) est compatible avec toutes les contraintes en $[X]$, y compris la contrainte libérée.

En effet, si nous reprenons les notations précédentes et si nous cherchons une direction $[D']$ telle que $\rho_{k_2} = 0$, cette direction sera compatible si la variable d'écart $z_{k_2} > 0$, toutes les autres variables d'écart restent nulles puisque nous conservons toutes les contraintes saturées en $[X]$ autres que celle d'indice k_2 .

Or, nous avons établi la relation (II, 20) :

$$[B] \{ [\Lambda] - [\rho] \} = [Z]$$

Dans ce cas, $[Z]$ a un seul élément non nul Z_{k_2} (II, 20) entraîne par multiplication par $\{ [\Lambda] - [\rho] \}^T$:

$$\{ [\Lambda] - [\rho] \}^T [B] \{ [\Lambda] - [\rho] \}^T = \{ [\Lambda] - [\rho] \}^T [Z]$$

puisque $[B]$ est une matrice symétrique, définie, positive

$$\begin{aligned} \Rightarrow \{ [\Lambda] - [\rho] \}^T [Z] &\geq 0 \quad \forall [\rho] \\ \Rightarrow (\lambda_{k_2} - \rho_{k_2}) Z_{k_2} &\geq 0 \end{aligned} \quad (II, 25)$$

Si nous choisissons Z_{k_2} pour que $\rho_{k_2} = 0$, nous avons donc nécessairement :

$$\lambda_{k_2} Z_{k_2} \geq 0 \Rightarrow Z_{k_2} \geq 0 \quad \text{car} \quad \lambda_{k_2} > 0$$

Ce qui montre que la direction $[D']$ pour laquelle $\rho_{k_2} = 0$ est compatible avec l'inégalité :

$$[D']^T [\psi_{k_2}] = Z_{k_2} \geq 0 \quad (II, 26)$$

Puisque tous les autres éléments de $[Z]$ sont nuls, nous avons :

$$[D']^T [\psi_{i'}] = 0 \quad i' \in L \cup K' \quad (II, 27)$$

Or ces relations (II, 27) entraînent que $\|D'\|$ soit minimum (II-1) et que par conséquent les ρ_i^1 soient pseudo-solution de :

$$\text{grad } f([X]) - \sum_{i \in L \cup K} \rho_i^1 \psi_i = 0 \quad \text{grad } \psi_{i'} = 0 \quad (II, 28)$$

III - PLAN DE RECHERCHE DU MAXIMUM SOUS CONTRAINTES

Considérons le problème défini par (II, 1), (II, 2), (II, 3) :

$$\text{Max } [f([X]) / [X] \in (D)]$$

où (D) = domaine de $\mathbb{R}^n$ défini par $\begin{cases} \psi_i([X]) = 0 & i = 1, 2, \dots, p \\ \psi_j([X]) \geq 0 & j = p+1, \dots, q \end{cases}$

La technique proposée est une technique itérative mixte. Au cours du traitement, il se peut qu'au point $[X]$, certaines contraintes deviennent

saturées. Nous utilisons les mêmes notations qu'au paragraphe (II) pour distinguer les liaisons, les contraintes saturées et non saturées.

$$\begin{aligned} \varphi_{\ell}([X^r]) &= 0 & \ell \in L \text{ (liaisons)} \\ \varphi_k([X^r]) &= 0 & k \in K \text{ (contraintes saturées)} \\ \varphi_h([X^r]) &> 0 & h \in H \text{ (contraintes non saturées)} \end{aligned}$$

Donc :

$$\text{Card}(L) = p$$

Posons $s = \text{Card}(L) + \text{Card}(K)$

$\Rightarrow \text{Card}(H) = q - s$ où q est le nombre total des liaisons et des contraintes. Il y a donc s variables liées ($s \leq n$) et $(n-s)$ variables libres.

Posons $[X^r] = ([Y^r], [Z^r])$
 où $[Y^r] = \left\{ \begin{array}{l} \text{variables liées} \end{array} \right\}$
 $[Z^r] = \left\{ \begin{array}{l} \text{variables libres} \end{array} \right\}$

La recherche du maximum se compose d'une suite d'étapes. Chaque étape est composée d'une suite de pas. Le début de chaque étape est caractérisé par un nouveau point $[X^{r+1}] \in (D)$ tel que $f([X^{r+1}]) > f([X^r])$. Pour la première étape, nous appliquons la méthode de quadratisation partielle.

Supposons que nous sommes à la $r^{\text{ème}}$ étape, les différentes phases essentielles d'une étape sont les suivantes :

1ère phase Rechercher des contraintes saturées et non saturées, d'où les variables libres et les variables liées :

$$[X^{r+1}] = ([Y^{r+1}], [Z^{r+1}])$$

2ème phase Faire un test sur un indicateur logique ^{pour savoir quelle est la} méthode qui est utilisée pendant cette étape. Puis avec cette méthode nous calculons la nouvelle valeur des variables libres $\Rightarrow [Z^{r+1}]$

3ème phase Calculer les variables liées $[Y^{r+1}]$ par la résolution du système des liaisons et des contraintes saturées :

$$\left. \begin{aligned} \varphi_{\ell}([Y], [Z^{r+1}]) &= 0 & \ell \in L \\ \varphi_k([Y], [Z^{r+1}]) &= 0 & k \in K \end{aligned} \right\} \Rightarrow [Y^{r+1}]$$

D'où le nouveau point $[X^{r+1}] = ([Y^{r+1}], [Z^{r+1}])$

4ème phase Tester pour savoir s'il y a une amélioration, c'est-à-dire si $f([X^{r+1}]) > f([X^r])$.

Si oui, passons à la phase suivante

Sinon, nous divisons le coefficient t de la direction de montée des variables libres par 2 et nous recommençons la deuxième phase pour calculer la nouvelle $[Z^{r+1}]$. Signalons que nous prenons la valeur initiale de $t = 1$.

5ème phase Vérifier si $[X^{r+1}] \in (D)$

Sinon, nous divisons le t par 2 et allons à la deuxième phase.

Dans le cas contraire, continuons. Notons qu'à partir de la phase suivante, nous sommes sûrs d'avoir un nouveau point $[X^{r+1}] \in (D)$ tel que $f([X^{r+1}]) > f([X^r])$.

6ème phase Calculer les coefficients de Lagrange et de Kühn et Tucker en ce nouveau point et puis le résidu du système (II, 23) de n équations à s variables (le résidu d'un tel système sera défini dans le paragraphe suivant.)

7ème phase Effectuer un test pour savoir si les conditions de Kühn et Tucker sont remplies, à savoir :

- i) la sous-positivité des coefficients de Kühn et Tucker
- ii) la compatibilité du système (II, 23)

Si ces conditions sont vérifiées, nous avons atteint le point maximum et la recherche est terminée.

Sinon, nous préparons l'étape suivante en distinguant deux cas

- 1) Si la condition i) est vérifiée, nous appliquons la méthode de quadratisation partielle à l'étape suivante en gardant saturées toutes les contraintes saturées.
- 2) La condition i) n'est pas vérifiée, c'est-à-dire qu'il existe au moins un $\lambda_k > 0$ ($k \in K$). Dans ce cas, nous cherchons le $\lambda_k > 0$ ($k \in K$) de plus grande valeur et nous libérons la contrainte correspondant à ce coefficient. Ensuite, nous passons à l'étape suivante en utilisant la méthode du gradient projeté exposé au paragraphe II, en calculant la direction de montée $[D']$ définie au (II-3).

V - TECHNIQUE DE PROGRAMMATION

Dans ce paragraphe, nous allons décrire les techniques numériques utilisées pour résoudre les différentes phases décrites précédemment.

Tout d'abord nous devons distinguer, d'une part les variables liées et les variables libres, d'autre part les contraintes saturées (liaisons comprises) et les contraintes non saturées. Pour cette distinction, deux tableaux logiques sont utilisés :

IVL (n) avec $IVL_i = 0$ pour variables libres

$IVL_i = 1$ " " liées

ILU (q) avec $ILU_i = 1$ pour contraintes saturées ou liaisons

$ILU_i = 0$ pour " " non saturées.

n = dimension de l'espace des $[X]$

q = nombre total des liaisons et des contraintes.

1ère phase L'écriture du programme de cette phase ne présente aucune difficulté. Les valeurs des contraintes sont calculées par le sous-programme CØSALIAI[II]. Dans ce sous-programme, seules les contraintes saturées, c'est-à-dire que celles correspondant aux $ILU_i = 1$ sont calculées. Donc, si l'on veut calculer la valeur de toutes les contraintes en un point quelconque, nous utilisons un tableau de dimension q dont tous les éléments sont égaux à l'unité. Puis la recherche des nouvelles contraintes saturées est faite en comparant la valeur des contraintes calculée par la procédure CØSALIAI avec une valeur EPS prise comme paramètre du programme. S'il existe s contraintes bilatérales (liaisons) et contraintes unilatérales saturées, alors nous prenons d'abord les s premières variables liées, puis nous faisons appel au sous-programme CØRBASE [II]. Cette procédure permet d'améliorer la base en introduisant les variables libres x_i dans la base dans le cas où il existe une contrainte saturée de la forme $ax_i + b \geq 0$.

2ème phase La nature des variables et des contraintes est connue, c'est-à-dire que les tableaux ILU (n) et ILV (q) sont calculés, nous calculons alors la nouvelle valeur des variables libres. Tout d'abord, nous testons sur le paramètre logique ND en vue de connaître la méthode utilisée :

ND = 1 $\iff$ méthode de quadratisation partielle

ND = 0 $\iff$ méthode du gradient projeté.

Ensuite, par l'une de ces deux techniques, nous calculons la direction de montée $[\Delta Z^r] \Rightarrow$ nouvelle valeur $[Z^{r+1}]$. Nous notons que la méthode de quadratisation partielle est appliquée pendant la première étape.

a) ND = 1 Nous calculons les tableaux $[A^r]$, $[B^r]$, le hessien et le gradient au point $[X^r]$ par rapport aux variables libres $[Z^r]$. Ces calculs se font par le sous-programme XMAB. Dans XMAB, seuls les coefficients relatifs aux $[Z^r]$ sont calculés les deux matrices $[A^r]$ et $[B^r]$ sont donc creuses.

Puis par le sous-programme DIRMON, nous calculons $[\Delta Z^r]$. Dans le cas où le DIRMON ne permet pas d'extraire un sous-espace de l'espace des variables libres dont le hessien correspondant est défini négatif, nous divisons le pas par 2. Le nombre maximum de divisions des pas est fixé à NPAMAX. Donc il y a éventuellement un test de comparaison de NPAS (nombre de fois de division) avec NPAMAX. Si NPAS = NPAMAX et l'extraction reste toujours impossible, alors nous arrêtons le programme avec positionnement d'un indicateur.

b) ND = 0 Dans ce cas, la direction $[\Delta Z^r]$ se calcule par la procédure GRADIEN par la formule (II, 24). Une fois que $[\Delta Z^r]$ est obtenue la nouvelle valeur des variables libres :

$$[Z^{r+1}] = [Z^r] - t [\Delta Z^r]$$

3ème phase Cette phase sert à calculer la nouvelle valeur des variables liées $[Y^{r+1}]$ en connaissant $[Z^{r+1}]$. Ces calculs sont faits par la résolution du système des liaisons et des contraintes saturées.

$$\varphi_j([Y], [Z^{r+1}]) = 0 \quad j \in L \cup K$$

Prenons la même notation que précédemment :

$$s = \text{Card}(L) + \text{Card}(K)$$

Donc en général, c'est un système non linéaire de s équations à s inconnues. Nous résolvons par la méthode de Newton. A chaque itération nous avons un système creux à résoudre :

$$\left[\frac{\partial \varphi_j ([Y], [Z^{r+1}])}{\partial Y} \right]_{j \in L \cup K} [\Delta Y] = - \left[\varphi_i ([Y], [Z^{r+1}]) \right] \quad (II, 29)$$

dans lequel, seules les lignes et les colonnes correspondant aux liaisons et aux contraintes saturées ($ILU_i = 1$) et les colonnes correspondant aux variables liées $[Y]$ ($IVL_i = 1$) sont prises en compte.

Ce système est résolu par le sous-programme RESYLIDV [II]. Cette procédure permet de résoudre simultanément n systèmes creux de s équations à s inconnues. La technique utilisée se compose en deux étapes :

- * Comprimer le système creux en un système normal
- * Puis résoudre ce système normal par le sous-programme de bibliothèque MCADIR.

Le nombre d'itérations de la méthode de Newton est limité à MAX. Si après MAX itérations, il n'y a pas de convergence, nous effectuons un changement de base suivi par un appel au sous-programme CØRBASE qui est mentionné ci-dessus par la raison que le jacobien du système (II, 29) doit être non nul.

Le calcul des deux matrices creuses $\left[\frac{\partial \varphi_i ([Y], [Z^{r+1}])}{\partial Y} \right]$ et $\varphi_i ([Y], [Z^{r+1}])$ se fait par les sous-programmes DEPCØSAL et CØSALIAI [II].

4ème phase et 5ème phase

Ces deux phases sont facilement programmées. La valeur des contraintes et de la fonction est calculée respectivement par les sous-programmes CØSALIAI et VALFØN. Le test pour savoir si le nouveau point $[X^{r+1}]$ appartient ou non au domaine (D) se fait toujours par la comparaison de la valeur des contraintes avec EPS.

6ème phase A ce stade, un nouveau point $[X^{r+1}] \in (D)$ tel que $f([X^{r+1}]) > f([X^r])$ est obtenu. Calculons maintenant les coefficients de Lagrange des contraintes bilatérales et de Kühn et Tucker des contraintes unilatérales en $[X^{r+1}]$ par la résolution du système (II, 23) :

$$\begin{bmatrix} \psi^{r+1} \\ \lambda^{r+1} \end{bmatrix} = \begin{bmatrix} G^{r+1} \end{bmatrix}$$

où :

$$[\psi^{r+1}] =$$

$$\begin{bmatrix} \frac{\partial \phi_1 ([X^{r+1}])}{\partial x_1} & \dots & \frac{\partial \phi_j ([X^{r+1}])}{\partial x_1} & \dots \\ \frac{\partial \phi_1 ([X^{r+1}])}{\partial x_2} & \dots & \frac{\partial \phi_j ([X^{r+1}])}{\partial x_2} & \dots \\ \vdots & & \vdots & \\ \frac{\partial \phi_1 ([X^{r+1}])}{\partial x_n} & \dots & \frac{\partial \phi_j ([X^{r+1}])}{\partial x_n} & \dots \end{bmatrix}$$

$$[\lambda^{r+1}] = \begin{bmatrix} \lambda_1^{r+1} \\ \vdots \\ \lambda_j^{r+1} \end{bmatrix}$$

$$[G^{r+1}] = \begin{bmatrix} \frac{\partial f ([X^{r+1}])}{\partial x_1} \\ \vdots \\ \frac{\partial f ([X^{r+1}])}{\partial x_n} \end{bmatrix}$$

avec $j \in L \cup K$

La matrice $[G^{r+1}]$ est calculée à partir de l'expression explicite du gradient par la procédure GRADSF. Nous résolvons (II, 32) par le sous-programme MCADIR, d'où $[\lambda^{r+1}]$ est calculée.

Posons maintenant $[C_1] = [\psi^{r+1}] [\lambda^{r+1}]$.

Le résidu du système (II, 23) est défini par :

$$R = \text{Max}_{i=1, 2, \dots, n} \left| C1_i - \frac{\partial f ([X^{r+1}])}{\partial x_i} \right|$$

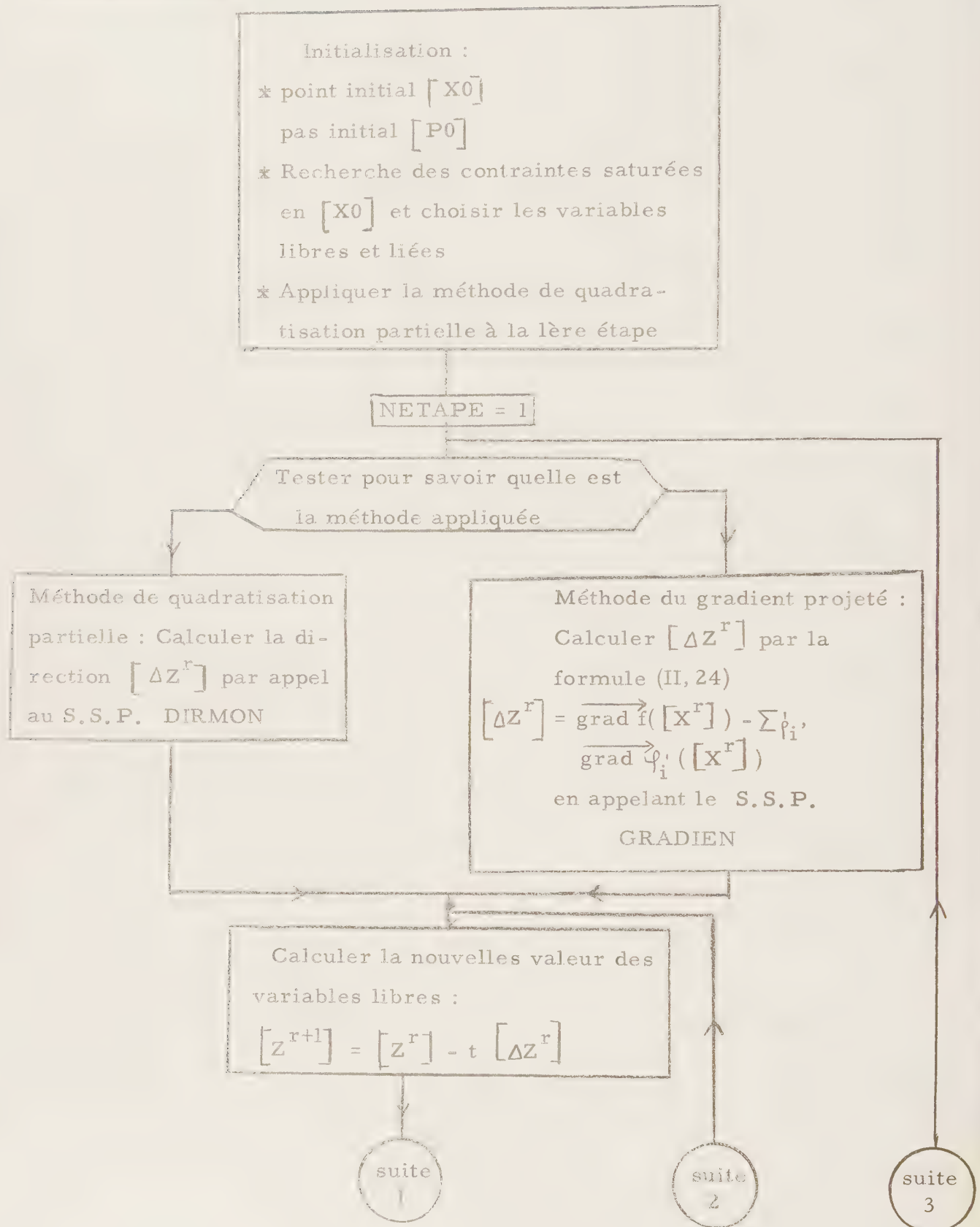
Les calculs du tableau $[\lambda^{r+1}]$ et du résidu R sont faits par le sous-programme REMAXCKT [II] qui, à partir du point $[X^{r+1}]$ donne $[\lambda^{r+1}]$ et R.

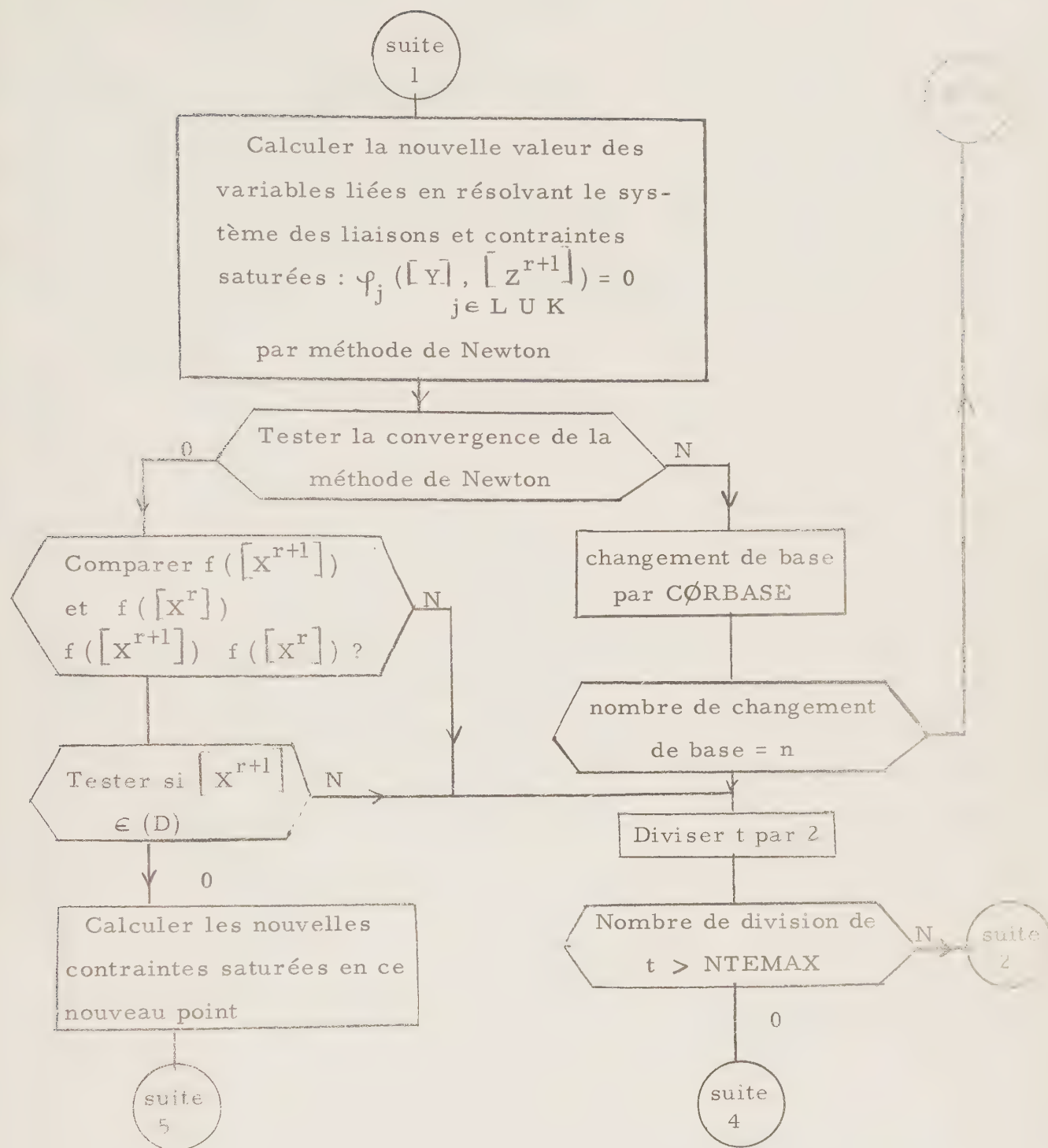
7ème phase Une fois que $[\lambda^{r+1}]$ et R ont été calculés, il ne reste qu'à vérifier les conditions de Kuhn et Tucker. Si ces conditions sont vérifiées, à savoir :

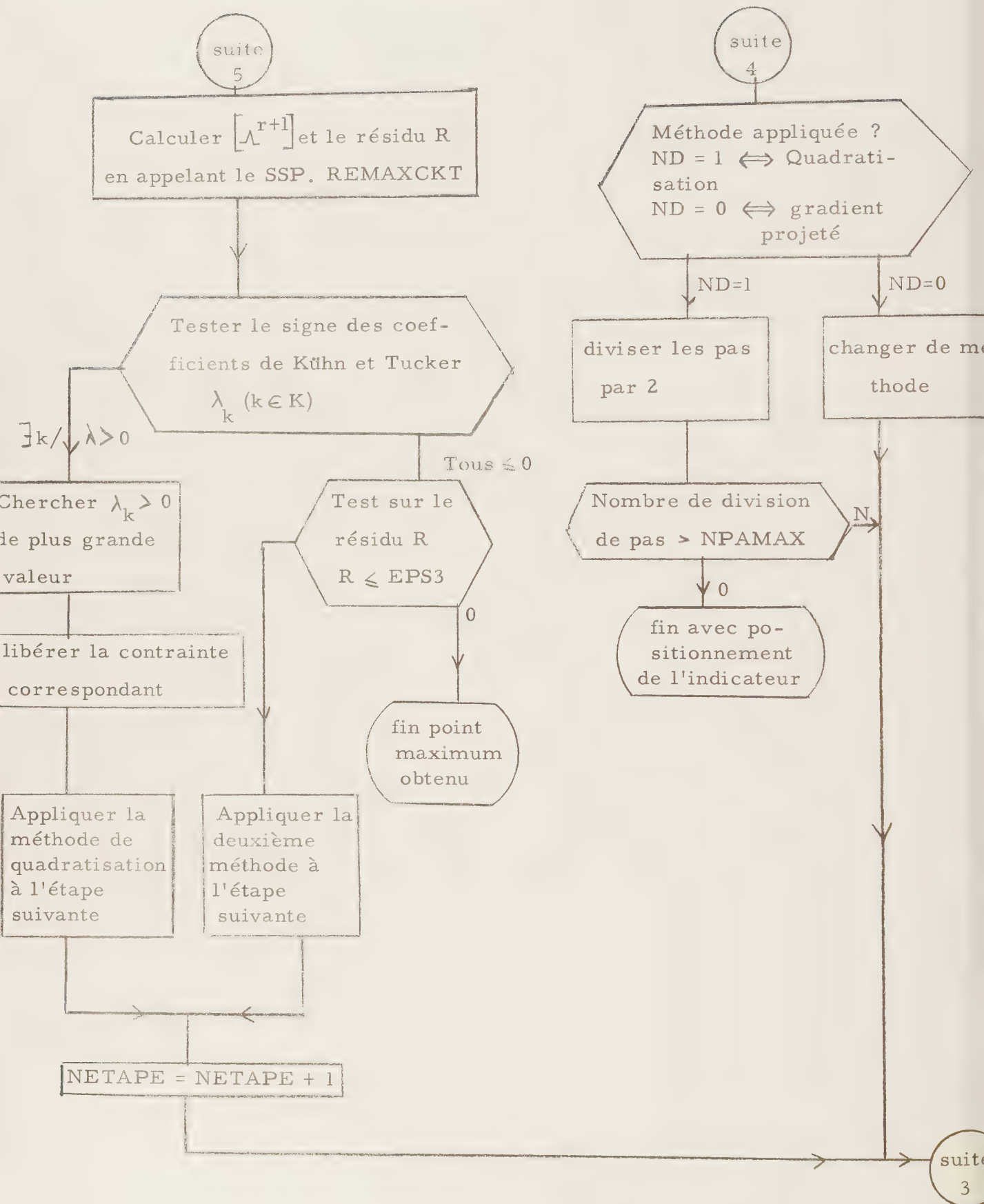
$$\begin{cases} \lambda_k \leq \text{EPS2} \\ \text{et } R \leq \text{EPS} \end{cases}$$

nous obtenons alors le point maximum. Sinon, nous recommençons le calcul en distinguant deux cas comme indiqué la III (7ème phase).

ORGANIGRAMME







III - RESULTATS DES ESSAIS NUMERIQUES

Des essais sont faits sur la fonction :

$$f([X]) = -x_1^2 - 10x_2^2 - x_3^2 - e^{x_2 x_3} - 100x_4^2 + x_5 - e^{x_5} + 2$$

avec les contraintes

$$\varphi_1([X]) = x_1 + x_2 + x_3 - x_4 e^{x_5} \quad 0$$

$$\varphi_2([X]) = x_3 + x_4 \quad 0$$

$$\varphi_3([X]) = x_1 - x_2^2 \quad 0$$

$$\varphi_4([X]) = 10 - \sum_{i=1}^5 x_i^2 \quad 0$$

$$\varphi_5([X]) = 3 - x_5 \quad 0$$

La solution de ce problème $M_0(0; 0; 0; 0; 0)$; $f(M_0) = 0$. Nous avons effectué deux essais, l'un à partir d'un point qui est à l'intérieur du domaine, l'autre avec un point sur la frontière du domaine. Les résultats sont respectivement dans les tableaux n° (10) et n° (11). Dans ces tableaux, la colonne "Méthode utilisée" nous indique effectivement la méthode utilisée pendant cette étape pour calculer les variables libres :

(I) $\iff$ méthode de quadratisation partielle

(II) $\iff$ méthode du gradient projeté.

A coté des colonnes des x_i , il y a les colonnes marquant la valeur des IVL_i :

$IVL_i = 0 \iff$ variable libre

$IVL_i = 1 \iff$ variable liée

Enfin, les colonnes des ILU_i se composent de deux petites colonnes qui indiquent respectivement la nature des contraintes avant et après les tests de Kuhn et Tucker.

$ILU_i = 0 \iff$ non saturée

$ILU_i = 1 \iff$ saturée.

TROISIEME PARTIE

APPLICATION A LA RESOLUTION
D'UN PROBLEME DE COMMANDE OPTIMALE

Dans cette partie, nous allons appliquer ces techniques pour résoudre un problème de commande optimale particulier. Il s'agit de la recherche des profils de température qui maximisent la production d'un corps chimique pendant un certain temps donné dans un réacteur tubulaire.

1 - RAPPEL DU PROBLEME DE COMMANDE OPTIMALE

Le comportement dynamique d'un système est décrit par un système d'équations différentielles :

$$\begin{cases} \dot{x}_1 = f_1(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m, t) \\ \dot{x}_2 = f_2(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m, t) \\ \vdots \\ \dot{x}_n = f_n(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m, t) \end{cases}$$

$$t_0 \leq t \leq t_f$$

avec les conditions initiales $x_i(t_0) = x_{i0}$ $i = 1, 2, \dots, n$. Sous forme matricielle, ce système s'écrit :

$$\left[\frac{dx}{dt} \right] = \left[f([x], [u], t) \right]$$

où :

$$[x] = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \left[\frac{dx}{dt} \right] = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix}, \quad [u] = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix},$$

$$\left[f([x], [u], t) \right] = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \quad [x(t_0)] = [x_0]$$

$[x]$ = matrice des variables d'état

$[u]$ = matrice des variables de commande

où les p_i , E_i sont des constantes appelées respectivement facteur de fréquence et énergie d'activation, et T est la température absolue à l'instant t .

Les valeurs numériques des p_i , E_i dans ce problème :

$$\begin{aligned} p_1 &= 0.535 \times 10^{11} \quad \text{min}^{-1} \\ p_2 &= 0.461 \times 10^{18} \quad \text{min}^{-1} \\ E_1 &= 18000 \quad \text{cal/mole} \\ E_2 &= 30000 \quad \text{cal/mole} \end{aligned}$$

Le problème consiste à trouver une fonction $T_M(t)$ qui maximise la concentration $x_2(t_1)$ du produit B où t_1 est l'instant final donné ($t_1 = 10 \text{ mm}$).

II - 2 Méthode de résolution utilisée

Nous remarquons que dans notre problème, il n'y a qu'une seule variable de commande $u(t) = T(t)$ et que la fonction de critère $S = x_2(t_1)$. Pour le résoudre ; nous approchons $T(t)$ par une fonction $f(t)$ de forme donnée en nous limitant aux formes suivantes :

$$\begin{aligned} * \quad T(t) &= a_1 \quad (\text{cas d'un processus isothermique}) \\ * \quad T(t) &= a_1 \exp(-a_2 t) \\ * \quad T(t) &= a_1 \exp(-a_2 t) + a_3 \end{aligned}$$

Le problème revient donc à maximiser la fonction $x_2(t_1)$ par rapport aux coefficients inconnus a_1, a_2, a_3 .

Pour chaque valeur des (a_i) , nous calculons celle de la fonction de critère $x_2(t_1)$ en résolvant le système d'équations différentielles (III,1) et (III,2) avec les conditions initiales (III,3), (III,4). Cette résolution se fait par la méthode à pas liés implicite à deux termes.

Soit le système :

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \dots, x_n, t) \quad i = 1, 2, \dots, n$$

En divisant l'intervalle $[0, t_1]$ en p intervalles égaux et en intégrant cette relation entre t_{r-1} et t_r , nous avons :

$$\int_{t_{r-1}}^{t_r} dx_i = \int_{t_{r-1}}^{t_r} f_i(x_1, x_2, \dots, x_n, t) dt$$

$$\Rightarrow x_i^r - x_i^{r-1} = \frac{\Delta t}{2} \left[f_i(x_1^r, \dots, x_n^r, t_r) + f_i(x_1^{r-1}, \dots, x_n^{r-1}, t_{r-1}) \right]$$

$$\text{où } x_i^r = x_i(t_r)$$

Appliquons cette relation à notre problème, nous obtenons les relations suivantes :

$$x_1^r = \frac{2 - \Delta t \times k_1(t_{r-1})}{2 + \Delta t \times k_1(t_r)} x_1^{r-1}$$

$$x_2^r = \frac{2 - \Delta t \times k_2(t_{r-1})}{2 + \Delta t \times k_2(t_r)} x_2^{r-1} + \frac{\Delta t \times k_1(t_{r-1})}{2 + \Delta t \times k_2(t_r)} x_1^{r-1} + \frac{\Delta t \times k_1(t_r)}{2 + \Delta t \times k_2(t_r)} x_1^r$$

(III, 8)

Ces relations permettent de calculer $x_2(t_1)$.

II - 3 - Résultats numériques

II. 3.1 Méthode de quadratisation partielle

a) - Cas d'un processus isothermique $T(t) = a_1$.

NR	a_1	$x_2(t_1)$
0	330	0.5309
1	340.80	0.6716
2	339.83	0.6738
3	339.80	0.6738

b) Cas où $T(t) = a_1 \exp(-a_2 t)$

pas initial = (0.1 ; 0.1)

Résultat dans le tableau n° 9.

c) Cas où $T(t) = a_1 \exp(-a_2 t) + a_3$

pas initial = (0.1 ; 0.1 ; 0.1)

Résultat dans le tableau n° 10

NR.	a_1		a_2		$x_2(t_1)$	
0	330		0.00	1	0.5309	
1	343.49	1	-0.20×10^{-2}	0	0.5335	
2	335.04	1	-0.20×10^{-2}	1	0.6643	
3	337.49	1	-0.53×10^{-2}	1	0.6684	
5	341.39	1	0.10×10^{-2}	1	0.6762	
7	344.25	1	0.27×10^{-2}	1	0.6783	
9	345.08	1	0.32×10^{-2}	1	0.6784	
13	345.10		0.32×10^{-2}		0.6784	

Tableau n° 9

NR.	a_1		a_2		a_3	$x_2(t_1)$	
0	330	1	0.00	0	20.00	1	0.4266
1	318.90	0	0.00	1	8.90	0	0.4800
2	318.90	1	-0.76×10^{-2}	1	8.90	0	0.6090
3	315.74	1	-0.84 "	1	8.90	0	0.6227
4	315.97	1	-0.72 "	1	8.90	0	0.6255
5	316.88	1	-0.63 "	1	8.90	0	0.6391
7	322.78	1	-0.41 "	1	8.90	0	0.6564
9	322.88	1	-0.38 "	1	8.90	0	0.6568
11	322.28	1	-0.14 "	1	8.90	0	0.6693
14	331.66	1	0.63×10^{-3}	1	8.90	0	0.6752
18	335.99	1	0.32×10^{-2}	1	8.90	1	0.6784
20	335.92	1	0.33×10^{-2}	1	9.19	1	0.6784
23	335.91		0.33×10^{-2}		9.20		0.6784

Tableau n° 10

II.3.2 Méthode de quadratisation mixte

a) Pour $\lambda_0 = 2.5 \times 10^{-3}$

NR	a_1		a_2		a_3		$x_2(t_1)$
0	330.00	1	0.00	0	20.00	1	0.4266
1	324.45	1	0.11×10^{-2}	1	14.45 ^(*)	0	0.6618
2	328.44	1	0.24 "	1	14.45 ^(*)	0	0.6769
3	330.00	1	0.32 "	1	14.45 ^(*)	0	0.6782
4	330.72	1	0.35 "	1	14.45 ^(*)	0	0.6782
5	330.81	1	0.35 "	1	14.45 ^(*)	0	0.6784
6	330.80	1	0.35 "	1	14.45 ^(*)	0	0.6784
7	330.72	1	0.34 "	1	14.45 ^(*)	1	0.6784
9	330.06	1	0.34 "	1	14.49	1	0.6784
11	330.06		0.34×10^{-2}		14.49		0.6784

b) Pour $\lambda_0 = 5 \times 10^{-3}$

Nous obtenons le point optimum :

$$\begin{cases} a_{1,M} = 328.69 \\ a_{2,M} = 0.34 \times 10^{-2} \\ a_{3,M} = 16.41 \end{cases} \Rightarrow x_2(t_1) = 0.6784$$

c) Pour $\lambda_0 = 1.25 \times 10^{-2}$

Nous obtenons le point optimum :

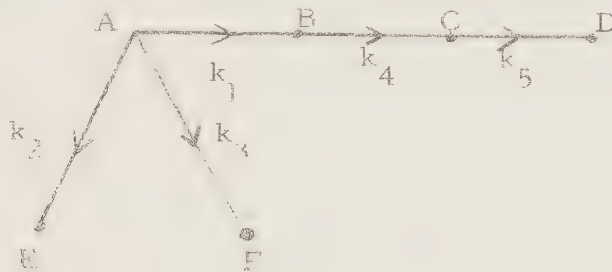
$$\begin{cases} a_{1,M} = 330.06 \\ a_{2,M} = 0.34 \times 10^{-2} \\ a_{3,M} = 14.50 \end{cases} \Rightarrow x_2(t_1) = 0.6784$$

(*) Ces valeurs ne se différent qu'à partir du troisième chiffre décimal

III - PROBLEME AVEC CONTRAINTES

C'est un problème analogue avec le problème précédent mais avec des contraintes dont la forme est simple.

Le schéma de la réaction est le suivant :



Soient x_1, x_2, x_3 les concentrations des produits A, B, C. Les équations différentielles régissant cette réaction :

$$\begin{cases} \frac{dx_1}{dt} = -k_1 x_1 - k_2 x_1 - k_3 x_1 \\ \frac{dx_2}{dt} = k_1 x_1 - k_4 x_2 \\ \frac{dx_3}{dt} = k_4 x_2 - k_5 x_3 \end{cases}$$

Les k_i sont définis par :

$$k_i = C_i \exp \left[- \frac{E_i}{R} \left(\frac{1}{T} - \frac{1}{658} \right) \right]$$

où C_i, E_i

i	1	2	3	4	5
C_i	1.02	0.93	0.386	3.28	0.084
E_i	16000	14000	15000	10000	15000

Le problème est de chercher $T(t)$ qui maximise $x_3(t_1)$ avec la condition :

$$T(t) \leq 823^\circ \text{ K}$$

(contrainte imposée par des considérations chimiques).

Le système des équations différentielles est résolu par discrétisation comme dans le paragraphe précédent.

Nous cherchons $T(t)$ de la forme :

$$T(t) = a_1 \exp(-a_2 t).$$

D'où nous avons les contraintes :

$$\left\{ \begin{array}{l} a_1 \geq 0 \\ 823 - a_1 \geq 0 \\ a_2 \geq 0 \end{array} \right. \quad (\text{en supposant que } T(t) \geq 0)$$

Deux essais ont été effectués.

Résultats

1er essai : pas initial = (0.1 ; 0.1)

NETAPE	a_1	a_2	$x_3(t_1)$
0	750	0.05	0.3974
1	797.46	0.3088	0.4149
2	815.86	0.4277	0.4206
3	822.55	0.4729	0.4226
4	822.93	0.4755	0.4228
5	823.00	0.4755	0.4228
6	823.00	0.3888	0.4255
7	823.00	0.3940	0.4255
8	823.00	0.3927	0.4255

2ème essai : pas initial = (0.1 ; 0.1)

NETAPE	s_1	a_2	$x_3(t_1)$
0	800	0.5	0.3902
1	818.43	0.4657	0.4199
2	821.04	0.4833	0.4208
3	822.33	0.4920	0.4211
4	822.97	0.4964	0.4213
5	822.99	0.4965	0.4213
6	823.00	0.4966	0.4213
7	823.00	0.4966	0.4213
8	823.00	0.3846	0.4255
9	823.00	0.3940	0.4255
10	823.00	0.3927	0.4255

Remarques

Nous notons que dans les essais du problème avec contraintes :

i) la formule

$$\frac{f(x_1, \dots, x_i + h_i, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{h_i}$$

est utilisée pour calculer approximativement le gradient dans la
procédure GRADIEN

ii) Il n'y a que la méthode de quadratisation partielle qui est utilisée
car le coefficient de Kühn et Tucker de la contrainte saturée reste
toujours négatif au cours du traitement.

IV - CONCLUSION

Nous avons transformé un problème de contrôle optimal où il n'y a qu'une seule variable de commande :

$$\text{où } \begin{cases} S = G ([\bar{x}], u(t), t) \\ \left[\frac{dx}{dt} \right] = [F ([\bar{x}], u(t), t)] \\ u(t) \in U \end{cases} \quad (\text{III}, 9)$$

en un problème d'optimisation statique en approchant la fonction de commande $u(t)$ par une fonction $f(a_1, a_2, \dots, a_n, t)$, le problème (III, 9) devient :

$$\text{où } \begin{cases} S = H ([\bar{x}], [a], t) \\ \left[\frac{dx}{dt} \right] = K ([\bar{x}], [a], t) \\ a = \text{matrice des coefficients } a_i \in A \end{cases}$$

et ensuite utilisée les techniques exposées précédemment pour résoudre ce dernier.

Les résultats ci-dessus montrent que ces procédés paraissent parfaitement applicables pour résoudre certains problèmes d'optimisation dynamique.

PROGRAMMES FORTRAN
.....

NOTICES D'UTILISATION
.....

Nous donnons dans cette partie les sous-programmes des techniques exposées précédemment. Ils sont écrits en FORTRAN IV et ont été mis au point sur CII 10070. Les cartes de commande sont écrites dans le système SIRIS 7

I - METHODE DE QUADRATISATION PAR INTERPOLATION

I-1 Sous-programme QUADRIN

```

1:      SUBROUTINE QUADRIN(X0,P0,X1,FM1,INDI,AL,Y,SUM,IP,A,B,P1)
2:      DIMENSION XO(N),PO(N),X1(N),AL(N),Y(N),SUM(N),IP(N)
3:      1,A(N,N),B(N,1),P1(N)
4:      COMMON/NGN/N,NBMAX,NPAMAX/XGN/EP5,EP51
5:      NR =1
6:      DO 1 I=1,N
7:      1 P1(I)=PO(I)
8:      CALL VAL(FM0,X0,N)
9:      403 NPAS=1
10:
11:      C      CALCUL DES MATRICES A ET B
12:      C
13:      402 DO 3 I=1,N
14:      XO(I)=X0(I)+P1(I)
15:      CALL VAL(FI,X0,N)
16:      XO(I)=X0(I)-2.*P1(I)
17:      CALL VAL(FII,X0,N)
18:      A(I,I)= (FI+FII-2.*FM0)/(P1(I)*P1(I))
19:      XO(I)=X0(I)+P1(I)
20:      3 B(I,1)= -(FI-FII)/(2.*P1(I))
21:      DO 55 I=1,N-1
22:      XO(I)=X0(I)+P1(I)
23:      CALL VAL(FI,X0,N)
24:      XO(I)=X0(I)-P1(I)
25:      DO 13 J=I+1,N
26:      XO(J)=X0(J)+P1(I)
27:      CALL VAL(FJ,X0,N)
28:      XO(I)=X0(I)+P1(I)
29:      CALL VAL(FIJ,X0,N)
30:      XO(I)=X0(I)-P1(I)
31:      XO(J)=X0(J)-P1(J)
32:      13 A(I,J)= (FIJ-FI-FJ-FM0)/(P1(I)*P1(J))
33:      55 CONTINUE
34:      DO 7 J =2,N
35:      DO 7 I =1,J-1
36:      7 A(J,I)=A(I,J)
37:
38:      C      CALCUL DU NOUVEAU POINT
39:      C
40:      CALL MCADIR(A,B,N,N,N,N,1,1,AL,Y,SUM,IP)
41:      IF(IP(1).NE.0) GOTO 31
42:      DO 9 I =1,N
43:      9 X1(I)=X0(I)+B(I,1)
44:      CALL VAL(FM1,X1,N)

```

```

431 C
441 C TEST DE SORTIE
451 C
461 IF (FM1-FM0) 30,30,31
471 30 D9 40 I=1,N
481 40 P1(I)=P1(I)/2.
491 NBAS=NBAS+1
501
511 IF (FM1-FM0) 40,40,41 GBT9 2000
521 GBT9 402
531 41 NR =NR+1
541 IF (NR.GT.NBMAX) GBT9 2001
551 D9 302 I =1,N
561 302 X0(I)=X1(I)
571 ZZ=FM0
581 FM0=FM1
591 IF (ABS(FM1-ZZ )-EPS) 10,10,11
601 11 D9 41 I=1,N
611 41 P1(I)=P0(I)
621 GBT9 403
631 10 D9 610 I =1,N
641 X0(I)=X0(I)+P1(I)
651 CALL VAL(FI,X0,N)
661 X0(I)=X0(I)-2.*P1(I)
671 CALL VAL(FII,X0,N)
681 X0(I)=X0(I)+P1(I)
691 610 B(I,1)=- (FI-FII)/(2.*P1(I))
701 D9 500 I = 1,N
711 IF (ABS(B(1,1))-EPS1) 500,500,501
721 500 CONTINUE
731 GBT9 4000
741 501 D9 307 I =1,N
751 307 P1(I)=P1(I)/2.
761 GBT9 402
771
781 C
791 C ECRIRE LES RESULTATS
801 C
811 4000 INDI=0
821 RETURN
831 2000 INDI=1
841 RETURN
851 3001 INDI=2
861 RETURN
871 END

```


I - 2 Notice d'utilisation

Ce sous programme recherche un minimum local par la méthode de quadratisation par interpolation.

1) Utilisation

CALL QUADRIN (X0, P0, X1, FM1, INDI, AL, Y, SUM, IP, A, B, P1)

2) Description des paramètres et arguments formels

N : dimension de l'espace $\mathbb{R}^n$ des variables

NOMAX : nombre maximum d'étapes

NPAMAX : nombre maximum de fois de division des pas par 2

EPS : précision pour la comparaison des valeurs de la fonction aux points $[X^r]$ et $[X^{r+1}]$

EPS1 : précision pour la sortie

$$\|B([X^r])\| \leq \text{EPS1} \quad (\|B([X^r])\| = \text{Max}_i |b_i([X^r])|)$$

XO : tableau de N éléments définissant le point initial

PO : tableau de N éléments définissant le pas initial

X1 : tableau de N éléments définissant le point minimum

FM1 : valeur de la fonction au point minimum

INDI : indicateur de condition d'arrêt du programme

* INDI = 0 $\iff$ point minimum est atteint

* INDI = 1 $\iff$ nombre de division des pas > NPAMAX

* INDI = 2 $\iff$ nombre d'étapes > NOMAX

3) Tableaux de travail

A : tableau de N lignes et N colonnes approchant le hessien

B : tableau de N éléments approchant l'opposé du gradient

AL, Y, SUM, IP, P1 : Tableaux unidimensionnels de N éléments.

4) Sous-programmes utilisés

a) Sous-programme MCADIR (procédure existante de la bibliothèque de l'Institut Universitaire de Calcul Automatique de Nancy)

```

1: SUBROUTINE MCADIR(QR,R,M,MN,N,NN,P,PN,AL,Y,SUM,IP)
2: INTEGER PN
3: INTEGER P
4: DIMENSION QR(MN,NN),R(MN,PN),AL(NN),IP(NN),SUM(MN),Y(MN)
5: DO 10 J=1,N; D=0; DO 11 I=1,M; QA=QR(I,J)
6: 1 D=D+QA*QA; SUM(J)=D
7: IP(J)=J; DO 20 K=1,N; SIG=SUM(K); JBAR=K; IF(K.EQ.N) GO TO 75
8: DO 13 J=K+1,N; IF(SIG-SUM(J)) 14,13,13
9: SIG=SUM(J); JBAR=J
10: CONTINUE
11: IF(JBAR.EQ.K) GO TO 15
12: I=IP(K); IP(K)=IP(JBAR); IP(JBAR)=I; SUM(JBAR)=SUM(K); SUM(K)=
13: DO 16 I=1,M; SIG=QR(I,K); QR(I,K)=QR(I,JBAR)
14: QR(I,JBAR)=SIG
15: D=0; DO 2 I=K,M; QA=QR(I,K)
16: D=D+QA*QA; SIG=D; IBAR=K; IF(SIG.LE.1E-74)GO TO 33
17: QRKK=QR(K,K); ALK=SQRT(SIG); IF(QRKK.GE.0) ALK=-ALK
18: AL(K)=ALK; BETA=1/(SIG-QRKK*ALK); QR(K,K)=QRKK-ALK; IF(K.EQ.N)GO
19: DO 18 J=K+1,N; D=0; DO 3 I=K,M
20: D=D+QR(I,K)*QR(I,J)
21: Y(J)=BETA*D; DO 20 J=K+1,N; DO 21 I=K,M
22: QR(I,J)=QR(I,J)-QR(I,K)*Y(J)
23: SUM(J)=SUM(J)-QR(K,J)*QR(K,J)
24: DO 55 K=1,P; DO 56 I=1,M
25: 56 Y(I)=R(I,K); DO 50 J=1,N; GA=0; DO 51 I=J,M
26: 51 GA=GA+Y(I)*QR(I,J); GA=GA/(AL(J)*QR(I,J)); DO 50 I=J,M
27: 50 Y(I)=Y(I)+GA*QR(I,J); SUM(N)=Y(N)/AL(N); DO 52 K1=1,N-1; I=N-K1
28: R3=-Y(I); DO 53 J=I+1,N
29: R3=R3+SUM(J)*QR(I,J)
30: SUM(I)=-R3/AL(I); DO 55 I=1,N; IRA=IP(I)
31: R(IRA,K)=SUM(I)
32: IP(1)=0
33: RETURN
34: IP(1)=JBAR
35: RETURN
36: END

```

b) Sous-programme VAL(FI,X0,N) définissant la valeur de la fonction au point X0.

FI : valeur de la fonction au point X0

XO : tableau de N éléments définissant le point considéré

N : dimension du tableau X0.

5) Exemple d'utilisation

JBR,B MALEGR15-A9C,IUCA,NGUYEN
 LIMIT (TIME,7),(C9RE,44),(SPDISC,40)
 FORTRAN SI,LS,G9,ADP,NS

```

1: SUBROUTINE VAL(FI,X,N)
2: DIMENSION X(1)
3: DIMENSION Y(30)
4: DO 1 J=1,30
5:   Y(J)=(J-1)/29.
6:   F1=(Y(J)-X(1)*X(1))-1.0+2.0*X(1)*X(1)
7:   DO 3 I=2,30
8:     S=T=0.
9:     T=X(1)
10:    DO 2 J=2,6
11:      S=S+(J-1)*X(J)*Y(I)**(J-2)
12:      T=T+X(J)*Y(I)**(J-1)
13:    F1=F1-(S+T*T-1.0)**2
14:    F1=-F1
15:    RETURN
16:  END

17: C9MMEN/NGN/N,N9MAX,NPAMAX/XGN/EP5,EP51
18: DIMENSION X0(20),PO(20),X1(20),AL(20),Y(20),SUM(20),IP(20)
19: 1,A(20,20),B(20,1),P1(20)
20: N=6 ; EP5=0.1E-03 ; EP51 =0.1E-05
21: N9MAX=100 ; NPAMAX=10
22: DO 5 I1 = 1,10
23:   READ(105,101)(X0(I),I=1,N)
24:   READ(105,101)(PO(I),I=1,N)
25: 101 FORMAT(10F6.6,10F6.6,10F6.6,10F6.6,10F6.6,10F6.6)
26:   WRITE(108,102)
27: 102 FORMAT(1H0,3X,'POINT INITIAL'/)
28:   OUTPUT,(X0(I),I=1,N)
29:   + 10F(10*F6.6)
30: 103 FORMAT(1H0,3X,'PAS INITIAL'/)
31:   OUTPUT,(PO(I),I=1,N)
32:   CALL QJADRIN(X0,PO,X1,FM1,INDI,AL,Y,SUM,IP,A,B,P1)
33:   WRITE(108,106)
34: 104 FORMAT(1H0,3X,'VALEUR DE L INDICATEUR INDI'/)
35:   OUTPUT,INDI
36:   WRITE(108,104)
37: 105 FORMAT(1H0,3X,'POINT OPTIMUM'/)
38:   OUTPUT,(X1(I),I=1,N)
39:   WRITE(108,105)
40: 106 FORMAT(1H0,3X,'VALEUR DE LA FONCTION AU POINT OPTIMUM'/)
41:   OUTPUT,FM1
42: 5 CONTINUE
43: STOP
44: END

45: ASSIGN ETI,FIL,(SYS,BLD),(NAM,F4L1B),(UNT,AC,1SV9)
46: LINK

47: OPTION (UNSAT,ETI)
48: RUN

```

Résultats :

O PRINT INITIAL

X2(I) = .1000000000000000
 X0(I) = 1.500000000000000
 X0(I) = .000000000000000
 X2(I) = 1.500000000000000
 X0(I) = -1.000000000000000
 X0(I) = .500000000000000

O PAS INITIAL

P0(I) = .500000000000000
 P0(I) = .500000000000000
 P0(I) = .500000000000000
 P0(I) = .500000000000000
 P0(I) = .500000000000000
 P0(I) = .500000000000000

O VALEUR DE L INDICATEUR INDI

INDI = 0

O PRINT OPTIMUN

X1(I) = -1.572551149938272E-02
 X1(I) = 1.01243444450922
 X1(I) = -.232987149992679
 X1(I) = 1.26041460260366
 X1(I) = -1.51371008477578
 X1(I) = .992988069460850

O VALEUR DE LA FONCTION AU POINT OPTIMUN

FM1 = 2.287670054518438E-03

II - METHODE DE QUADRATISATION PARTIELLE

II - 1 Sous-programme QUAPAR

Cette procédure calcule le minimum d'une fonction par la méthode de quadratisation partielle.

```

1:      SUBROUTINE QUAPAR(X0,P0,X1,FM1,INDI,A,B,KRE,P1,X)
2:      DIMENSION X0(N),P0(N),X1(N),A(N,1),B(N,1),KRE(1),P1(1),X(1)
3:      COMMON /NGN/N,NBMAX,NTEMAX,NPAMAX,XNG,TE0,EPS,EPS1
4:      DO 1 I=1,N
5:      1 P1(I)=P0(I)
6:      CALL VAL(FM0,X0,N)
7:      NR=1
8:      403 NPAS= 1
9:      C
10:     C   CALCUL DES MATRICES A ET B
11:     C
12:     402 DO 3 I=1,N
13:         X0(I)=X0(I)+P1(I)
14:         CALL VAL(FI,X0,N)
15:         X0(I)=X0(I)-2.*P1(I)
16:         CALL VAL(FII,X0,N)
17:         A(I,I)= (FI+FII-2.*FM0)/(P1(I)*P1(I))
18:         X0(I)=X0(I)+P1(I)
19:         3 B(I,1)=- (FI-FII)/(2.*P1(I))
20:         DO 55 I=1,N-1
21:             X0(I)=X0(I)+P1(I)
22:             CALL VAL(FI,X0,N)
23:             X0(I)=X0(I)-P1(I)
24:             DO 13 J=I+1,N
25:                 X0(J)=X0(J)+P1(J)
26:                 CALL VAL(FJ,X0,N)
27:                 X0(I)=X0(I)+P1(I)
28:                 CALL VAL(FIJ,X0,N)
29:                 X0(I)=X0(I)-P1(I)
30:                 X0(J)=X0(J)-P1(J)
31:                 13 A(I,J)=(FIJ-FI-FJ+FM0)/(P1(I)*P1(J))
32:             55 CONTINUE
33:             DO 7 J =2,N
34:                 DO 7 I =1,J-1
35:                     7 A(J,I)=A(I,J)
36:             TE = TE0
37:     C
38:     C   EXTRAIRE DE A(N,N) UNE MATRICE DEFINIE POSITIVE
39:     C
40:         III=1
41:         34 IF(A(III,III))30,30,31
42:         30 DO 32 K<K=1,N
43:             A(III,K<K)=0.
44:         32 A(K<K,III)=0.
45:         KRE(III)=0
46:         III=III+1
47:         IF(III=N)100,100,101
48:     100 GET034
49:         31 A(III,III)=DSQRT(A(III,III))
50:         KRE(III)=1
51:         IF(III=N)200,71,71

```

```

52:      200 D=21 J = III+1,N
53:      21 A(J,III)=A(J,III)/A(III,III)
54:      D= 10 J=III+1,N
55:      S=0.
56:      D=23 I=1,J-1
57:      23 S=S+A(J,I)+A(J,I)
58:      S=A(J,J)-S
59:      IF(S)5,5,6
60:      5 KRE(J)=0
61:      D= 4 K=1,J
62:      4 A(J,K)=0.
63:      IF(J-N)70,71,71
64:      70 D=66 K=J+1,N
65:      66 A(K,J)=0.
66:      GOTO 10
67:      6 KRE(J)=1
68:      A(J,J)=DSQRT(S)
69:      IF(J-N)80,71,71
70:      *C D=27 I=J+1,N
71:      S=0.
72:      D= 8 K=1,J-1
73:      8 S=S+A(I,K)+A(J,K)
74:      S=A(I,J)-S
75:      27 A(I,J)=S/A(J,J)
76:      10 CONTINUE
77:      C
78:      C      DESUDRE R * TRANSP0SEE( R ) *B
79:      C
80:      71 IG=1
81:      +2 IF(      KRE(IG)) 40,41,40
82:      41 X(IG)=0.;B(IG,1)=0.;IG=IG+1
83:      GOTO 42
84:      40 X(IG)=B(IG,1)/A(IG,IG)
85:      B(IG,1)=X(IG)
86:      IF(IG-N)90,91,91
87:      90 D= 17 I =IG+1,N
88:      IF(      KRE(I)) 15,16,15
89:      16 X(I)=0.
90:      B(I,1)=0.
91:      GOTO 17
92:      15 TX=0.
93:      D= 12 J=1,I-1
94:      12 TX=TX+A(I,J)*X(J)
95:      X(I)=(B(I,1)-TX)/A(I,I)
96:      B(I,1)=X(I)
97:      17 CONTINUE
98:      91 J3=N
99:      52 IF(      KRE(JG)) 50,51,50
100:      51 X(JG)=0.;B(JG,1)=0.;JG=JG+1
101:      GOTO 52
102:      50 X(JG)=B(JG,1)/A(JG,JG)

```



```

103:         IF(JG=1)60,61,60
104:     60 D9 20 K=N-JG+2,N
105:         I=(N+1)-K
106:         IF( KRE(I)) 18,19,18
107:     19 X(I)=0.
108:         B(I,1)=0
109:         GOTO 20
110:     18 TX=0.
111:         D9 14 L=I+1,N
112:         J=N+(I+1)*L
113:     14 TX=TX+A(J,I)*X(J)
114:         X(I)=TX/(A(I,I)-TX/A(I,I))
115:     20 CONTINUE
116: C
117: C     CALCUL DU NOUVEAU POINT
118: C
119:         NTE = 1
120:     61 D9 62 I =1,N
121:         IF(KRE(I)*NE.1) GOTO 63
122:         X1(I) = XO(I)+TE*X(I)
123:         GOTO 62
124:     63 X1(I) = XO(I)
125:     62 CONTINUE
126:         CALL VAL(FM1,X1,N)
127: C
128: C     TESTS DE SORTIE
129: C
130:         IF(FM1-FM0)300,301,301
131:     301 NTE=NTE+1
132:         IF(NTE*GT*NTEMAX) GOTO 1000
133:         TE=TE/2.
134:         GOTO 61
135:     1000 NPAS=NPAS+1
136:         IF(NPAS*GT*NPAMAX) GOTO 2000
137:         D9 1002 I=1,N
138:     1002 P1(I)=P1(I)/2.
139:         GOTO 402
140:     300 NR=NR+1
141:         IF(NR*GT*NRMAX) GOTO 2001
142:         D9 302 I = 1,N
143:     302 XO(I) = X1(I)
144:         IF((FM0-FM1)-EPS)400,400,401
145:     401 D9 303 I =1,N
146:     303 P1(I) =P0(I)
147:         FM0 =FM1
148:         GOTO 403
149:     400 D9 610 I =1,N
150:         XO(I)=XO(I)+P1(I)
151:         CALL VAL(FI,XO,N)
152:         XO(I)=XO(I)-2.*P1(I)
153:         CALL VAL(FII,XO,N)

```

```

154:      X0(I)=X0(I)+P1(I)
155: 510 B(I,1)=- (F1-FI1)/(2.*P1(I))
156:      DO 500 I=1,N
157:      IF (ABS(B(I,1))-EPS1)500,500,501
158: 500 CONTINUE
159:      GOTO 502
160: 501 DO 307 I=1,N
161: 307 P1(I) =P1(I)/2.
162:      FMO=FM1
163:      GOTO 402
164: C
165: C      ECRIRE LES RESULTATS
166: C
167: 2000 INDI=1
168:      RETURN
169: 2001 INDI=2
170:      RETURN
171: 302 INDI=0
172:      RETURN
173: 101 INDI=3
174:      RETURN
175:      END

```

II - 2 Notice d'utilisation

1) Utilisation

CALL QUAPAR (X0, P0, X1, FM1, INDI, A, B, KRE, P1, X)

2) Description des paramètres et arguments formels

N, NOMAX, NPAMAX, EPS, EPS1, X0, P0, X1, FM1, A, B, P1 ont la même signification que dans le sous-programme QUADRIN

NTEMAX : nombre maximum de fois de division de t par 2

TEO : valeur initiale de t (dans les essais effectués dans ce travail, nous avons pris TEO = 1)

INDI : indicateur de condition d'arrêt du programme défini comme suit :

INDI = 0 == point minimum est obtenu

INDI = 1 == le pas devient trop petit et il n'y a pas encore de convergence (NPAS > NPAMAX)

$INDI = 2 \iff$ nombre d'étapes $NR > N\cancel{O}MAX$ et il n'y a pas encore de convergence
 $INDI = 3 \iff$ L'extraction d'une matrice $\left[A_p(X^r) \right]$ de l'approchant du hessien est impossible (Voir les notations de la page ())
 KRE tableau logique de N éléments
 $KRE_i = 1 \iff$ variable $x_i \in$ sous espace $\mathbb{R}^p$
 $KRE_i = 0 \iff$ variable $x_i \notin$ sous espace $\mathbb{R}^p$.

Nous rappelons que $\mathbb{R}^p$ est le sous-espace extrait de $\mathbb{R}^n$ tel que le hessien correspondant est strictement défini positif.

X : tableau de N éléments.

3) Sous programme utilisé

Il n'y a que le sous-programme VAL (FI, X0, N) décrit précédemment pour calculer la valeur de la fonction en un point.

4) Exemple d'utilisation

Nous prenons le même exemple que dans la méthode précédente, la procédure VAL (FI, X0, N) reste donc le même. Les résultats sont les mêmes.

III - RECHERCHE DU MINIMUM AVEC CONTRAINTES

III - 1 Sous programme QUACONT avec ses sous-programmes internes CORBASE [II] et REMAXCKT [II]

```

SUBROUTINE QUACONT(X0,P0,X1,FM,INDI,DI,CKT,A1,B1,A,B,C1,P1,H1,KRE)
DIMENSION X0(1),P0(1),X1(1),DI(1),CKT(NA,1),A1(NA,1),B1(NA,1),
1A(NA,1),B(NA,1),C1(1),P1(1),H1(1),KRE(1)
COMMON/NGN/IR,LP,N,NA,MAX,NPAMAX,NBMAX,NTEMAX,IVL,ILU,IVX,IP1,IP
1/XGN/TEC,EPS,EPS1,EPS2,EPS3,AL,Y,SUM
DIMENSION IVL(10),ILU(10),IVX(10),IP1(10),AL(10),Y(10),SUM(10)

```

```

C
C * SPPS PROGRAMME CALCULANT LE MAXIMISATION AVEC CONTRAINTES EN
C UTILISANT DEUX TECHNIQUES DE MONTÉE DECRIES DANS LA DEUXIEME
C PARTIE *
C
C

```

```

C * INITIALISATION ET PREPARATION DE LA PREMIERE ETAPE *
C

```

```

    NETAPE=1
    IP1=LP +1
    IS=LP
    DO 1 I =1,IR+N
1  IP1(I)=1
    DO 2 I=1,LP
2  ILU(I)=IVL(I)=1
    CALL COSALIAI(X0,C1,IP1)
    IF(LP.EQ.1R) GOTO 5
    DO 4 I=_LP1,1R
    IF(ABS(C1(I)).LT.EPS) GOTO 3
    ILU(I)=0
    GOTO 4
3  IS=IS+1
    IVL(IS)=ILU(I)=1
4  CONTINUE
5  DO 6 I =IS+1,N
6  IVL(I)=0

```

```

C
C * AMELIORER LA BASE SI C EST POSSIBLE PAR LE SSP CORBASE *
C

```

```

    CALL CORBASE
    FX0=G(X0)
    DO 7 I =1,N
    A1(I)=X0(I)
    B1(I)=P0(I)
    DO=1
5  NPAS=1
71 TF=TEC
    IF(ND.EQ.0) GOTO5008

```

```

C
C * CALCULER LES VARIABLES LIBRES , SOIT PAR LA METH DE QUADRATISATION
C PARTIELLE (ND=1) , SOIT PAR LA TECHNIQUE EXPOSEE AU (II-3) (ND=0) *
C

```

```

    CALL DIRMEN(DI,X1,A,B,NA,N,INDI,IP1,IVL,KRE)

```

```

C * INDI DEFINI COMME SUIT :

```

```
C * INDEX=0 EXTRACTION POSSIBLE DE LA METHODE DE QUADRATISATION PARTIELLE
C * INDEX=1 IMPOSSIBLE , ALORS DIVISONS LES PAS PAR DEUX
```

```

      IF (INDEX.EQ.1)      GOTO 26
      GOTO 13
5008 CALL GRADJEN(X1,C1,A1,IVL,ILU,NA,CKT,DI,IR,N)
      13 NTE=1
      130 DO 14 I =1,N
          X1(I) =XO(I)
          IF (IVL(I).EQ.0) X1(I)=XO(I)-TE*DI(I)
      14 CONTINUE

```

C CALCULER LES VARIABLES LIEES EN RESOLVANT LE SYSTEME DES LIAISONS
C ET DES CONTRAINTES SATUREES EN MO PAR LA METH DE NEWTON. LE NOMBRE
C D ITERATIONS TOLERABLES EST = MAX *

```

007007 I=1,N
007008 IF(ILU(I).NE.0) GOTO 7008
7007 CONTINUE
007009 GOTO 22
7008 D9 15 I =1,N
007010 IF(IVL(I).EQ.1)X1(I)=X0(I)
15 CONTINUE
007011 NITERA=0
16 CALL CUSALIAI(X1,C1,IP1)
007012 D9 17 I= 1,IR
007013 IF(ILU(I).EQ.1.AND.ABS(C1(I)).GT.EPS1)GOTO 18
17 CONTINUE
007014 NP=0
007015 INDIC2=0
007016 GOTO 22
18 D9 19 I=1,IR
19 B1(I,1)=C1(I)
007017 CALL DEPCUSAL(X1,A1,IVL,ILU,NA)
007018 CALL RESYLIDV(IVL,ILU,A1,B1,N,N,1,NA,AL,Y,SUM,IP,1)
007019 NITERA=NITERA+1
007020 IF(NITERA.GT.MAX)GOTO 21
007021 D9 20 I =1,N
007022 IF(IVL(I).EQ.1)X1(I)=X1(I)-B1(I,1)
20 CONTINUE
007023 GOTO 16

```

* SI APRES MAX ITERATIONS, PAS DE CONVERGENCE, NOUS CHANGONS LA BASE
PAR ROTATION CIRCULAIRE *

```

21 NP=NP+1
   IF(NP.GE.N) GOTO 26
   LG=IVL(N)
   DO 70 I = N,2,-1
70  IVL(I)=IVL(I-1)
   IVL(1)=LG

```

$$I_L(1) = -5$$

```

3:      D9 72 I=1,N
4:      X1(I)=X0(I)
5:      GOTO 71
6:
7:      SI NEWTON CONVERGE EN MOINS DE MAX ITERATIONS & CONTINUONS
8:
9:      82 FM1=G(X1)
10:
11:      C + TEST POUR S IL Y A AMELIORATION ET SI LE NOUVEAU POINT APPARTIENT
12:      C AU DOMAINE (DE) OU NON *
13:
14:      IF (FM1.GT.FM0) GOTO 24
15:
16:      C + SINON,REDUIRE LE COEFFICIENT TE ET RECOMMENCONS & ET PUIS REDUIRE
17:      C LES PAS *
18:
19:      26 NTE=NTE+1
20:      IF (NTE.GE.NTEMAX) GOTO 309
21:      TE=TE/2.
22:      GOTO 130
23:
24:      309 IF (ND.EQ.0) GOTO 301
25:      NPAS=NPAS+1
26:      IF (NPAS.GT.NPAMAX) GOTO 2002
27:      D9 23 I=1,N
28:      P1(I)=P1(I)/2.
29:      23 X1(I)=X0(I)
30:      GOTO 71
31:
32:      301 ND=1
33:      GOTO 71
34:
35:      24 CALL CSALIAI(X1,C1,IP1)
36:      D9 27 I=1,IR
37:      IF (C1(I).LT.(-EPS)) GOTO 26
38:
39:      27 CONTINUE
40:
41:      C
42:      C * RECHERCHER LES NOUVELLES CONTRAINTES SATUREES EN M1 *
43:      C
44:      JS=0
45:      IF (LP.EQ.1R) GOTO 30
46:      D9 29 I=1,IR
47:      IF (ILU(I).EQ.1) GOTO 29
48:      IF (ABS(C1(I)).GE.EPS) GOTO 29
49:      JS=JS+1
50:      IS=IS+1
51:      ILU(I)=1
52:      29 CONTINUE
53:
54:      30 IF (IS.LE.N) GOTO 31
55:      GOTO 2003
56:
57:      C
58:      C + CALCULER LES COEFFICIENTS DE KUNH-TUCKER PAR LE SSP REMAXCKT *
59:      C
60:      31 CALL REMAXCKT(X1)
61:      ND=1

```



```

C
C * TEST SUR LES CONDITIONS DE KUNH-TUCKER *
C
      D9 321=LP1,IR
      IF(ILU(I).EQ.0)      GOTO 32
      IF(CKT(I,1).LE.EPS2) GOTO 32
C
C * SINON IL Y A AU MOINS UN COEFFICIENT > 0 * ALORS CHERCHER LA
C   CONTRAINTE CORRESPONDANT AU COEFFICIENT >0 ET DE PLUS GRANDE
C   VALEUR * ENSUITE PREPARER L'ETAPE SUIVANTE EN LA LIBERANT *
C
      ND=0
      CKTMAX = CKT(I,1)
      J1=I
      ILU(J1)=0
      D9 5000 J=I+1,IR
      IF(ILU(J).EQ.0)      GOTO 5000
      IF(CKT(J,1).LT.CKTMAX) GOTO 5000
      CKTMAX=CKT(J,1)
      ILU(J1)=1
      ILU(J)=0
      J1=J
5000 CONTINUE
      IS=IS+1
      JS=JS+1
      GOTO 39
32 CONTINUE
C
C * ICI, TOUTES LES COEFFICIENTS SONT NON POSITIFS * FAIRE LE TEST SUR
C   LE RESIDU *
C
      IF(R.GT.EPS3) GOTO 39
C
C * SINON, LE MAXIMUM EST OBTENU *
C
      GO=GIX!!
      GOTO 2004
C
39 D9 40 I =1,N
      XO(I)=X1(I)
      P1(I)=P0(I)
      FMO=FM1
      NETAPE=NETAPE+1
      IF(NETAPE.GT.NOMAX) GOTO 2005
      D9 5002 I =1,IS
5002 IVL(I)=1
      D9 5003 I =IS+1,N
5003 IVL(I)=0
      GOTO 8
2003 INDI=1
      RETURN

```

```

2102 INDI=2
      RETURN
2103 INDI=3
      RETURN
2104 INDI=0

```

SUBROUTINE REMAXCKT(XZ)

* C9US PROGRAMME CALCULANT LES COEFFICIENT DE KUNH-TUCKER ET LE RESIDU

```

      DIMENSION XZ(1)
      R=0.
      CALL GRADSF(XZ,C1)
      IF (IS.EQ.0) GO TO 235
      DO 230 I=1,N
230  CKT(I,1)=C1(I)
      CALL DEPCBSAL(XZ,A1,IP1,ILU,NA)
      IM=0
      DO 232 I=1,IR
      IF (ILU(I) .EQ. 0) GO TO 232
      IM=IM+1
      DO 231 J=1,N
      A1(IM,J)=A1(I,J)
231  B1(J,IM)=A1(IM,J)
232  CONTINUE
      CALL MCADIR(B1,CKT,N,NA,IM,NA,1,1,AL,Y,SUM,IP)
      DO 234 I=1,N
      H1(I)=0.
      DO 233 J=1,IM
233  H1(I)=H1(I)+A1(J,I)*CKT(J,1)
      Q=ABS(C1(I)-H1(I))
      IF (R .LT. Q) R=Q
234  CONTINUE
      RETURN
235  DO 236 J=1,N
      Q=ABS(C1(I))
      IF (R .LT. Q) R=Q
236  CONTINUE
      RETURN

```

SUBROUTINE C9RBASE

* S9US PROGRAMME D AMELIORATION DE LA BASE *

```

      IJ=0
      DO 58 I=1,IR
      IF (IVX(I) .EQ. 0 .OR. ILU(I) .NE. 1) GO TO 58
      IF (IVL(IVX(I)) .EQ. 1) GO TO 58
74  IJ=IJ+1
      IF (IVL(IJ) .EQ. 0) GO TO 74
      DO 250 J=1+1,IR
      IF (IVX(J) .EQ. IJ .AND. ILU(J) .EQ. 1) GO TO 74
250  CONTINUE
      IVL(IJ)=0
      IVL(IVX(I))=1
58  CONTINUE
      RETURN
      END

```

1) Utilisation

CALL QUACONT (X0, P0, X1, FM, INDI, DI, CKT, A1, B1, A, B, C,
P1, H1, KRE)

2) Description des paramètres et arguments formels

IR : nombre total des contraintes
 LP : nombre total des contraintes bilatérales
 N : nombre de variables
 NA (entier) : $NA \geq \max (IR, N+2)$
 MAX : nombre maximum d'itérations toléré dans la résolution du système des contraintes saturées par la méthode de Newton.
 NPAMAX : nombre maximum de fois de division des pas par 2
 NØMAX : nombre maximum d'étapes
 NTEMAX : nombre maximum de fois de division de t par 2
 X0 : tableau de N éléments définissant le point initial ($X0 \in \text{domaine}(D)$)
 P0 : tableau de N éléments définissant le pas initial
 X1 : tableau de N éléments définissant le point optimum
 FM : valeur de la fonction au point optimum
 INDI : indicateur de condition d'arrêt du programme défini comme suit
 * $INDI = 0 \iff$ le point optimum est atteint
 * $INDI = 1 \iff$ nombre d'étapes $> NØMAX$
 * $INDI = 2 \iff$ nombre de division des pas $> NPAMAX$
 * $INDI = 3 \iff$ le nombre de variables liées devient supérieur au nombre total des variables.

3) Tableaux de travail

- DI, C1, P1, H1, AL, SUM : tableaux réels unidimensionnels de NA éléments
 - CKT, B : tableaux bidimensionnels de NA lignes et d'une colonne
 - KRE : tableau logique de N éléments
 - A, A1, B1 : tableaux bidimensionnels de NA lignes et de NA colonnes

- IVL, ILU, IVX, IP1, IP : tableaux entiers unidimensionnels de
NA éléments.

4) Sous-programmes utilisés

a) Sous programme XMAB calculant les approchants du hessien et du
gradient en un point. Dans ce sous-programme, nous ne calculons
que les lignes et les colonnes correspondant aux variables libres.

```

1:      SUBROUTINE XMAX(X0,P1,N,A,R,IVL,NA)
2:      DIMENSION X0(1),P1(1),IVL(1),A(NA,1),R(NA,1)
3:      FMO=G(X0)
4:      DO 2 I=1,N
5:      IF(IVL(I).EQ.1) GOTO 2
6:      X0(I)=X0(I)+P1(I)
7:      FI=G(X0)
8:      X0(I)=X0(I)-2.*P1(I)
9:      FII=G(X0)
10:     A(I,I)=((FI+FII)-2.*FMO)/(P1(I)*P1(I))
11:     X0(I)=X0(I)+P1(I)
12:     2 R(I,1)=-(FI-FII)/(2.*P1(I))
13:     DO 4 J=1,N-1
14:     IF(IVL(J).EQ.1) GOTO 4
15:     X0(J)=X0(J)+P1(J)
16:     FJ=G(X0)
17:     X0(J)=X0(J)-P1(J)
18:     DO 3 J=I+1,N
19:     IF(IVL(J).EQ.1) GOTO 4
20:     X0(J)=X0(J)+P1(J)
21:     FJ=G(X0)
22:     X0(J)=X0(J)+P1(J)
23:     FJU=G(X0)
24:     X0(J)=X0(J)-P1(J);X0(I)=X0(I)+P1(I)
25:     3 A(I,J)=-(FJ-FI-FJU+FMO)/(P1(I)*P1(J))
26:     4 CONTINUE
27:     DO 5 J=2,N
28:     DO 5 L=1,J-1
29:     5 A(J,I)=A(I,J)
30:     RETURN
31:     END

```

X0 : tableau définissant le point considéré.
P1 : tableau définissant le pas
A : tableau bidimensionnel de N lignes et de N colonnes
B : tableau bidimensionnel de N lignes et d'une colonne
IVL : tableau de N éléments défini comme suit :
$$IVL_i = 1 \Leftrightarrow \text{variable } x_i \text{ est de base}$$

$$IVL_i = 0 \Leftrightarrow \text{variable } x_i \text{ est hors-base}$$
N, NA : définis comme dans REMAX.

Dans ce sous-programme, nous utilisons le sous-programme
FUNCTION G(XA) pour calculer la valeur de la fonction en un point donné
XA.

b) Sous-programme FUNCTION G(XA) mentionné ci-dessus.

c) Sous-programmes RESYLIDV, DEPCØSAL, CØSALIAI, GRADSF, MCADIR

La description de ces sous-programmes sont faits dans le travail cité en [II] .

d) Sous-programme GRADIEN calculant la direction de montée des variables libres dans le cas où la méthode du gradient projeté est utilisée.

```
SUBROUTINE GRADIEN(X,C1,A1,IVL,ILU,NA,CKT,DIRECT,IR,N)
DIMENSION X(1),A1(NA,1),IVL(1),ILU(1),CKT(NA,1),DIRECT(1),C1(1)
CALL GRADSF(X,C1)
CALL DEPCØSAL(X,A1,IVL,ILU,NA)
DO 1 I = 1,N
IF (IVL(I) .NE. 1) GOTO 1
DO 2 J = 1,IR
S = 0.
IF (ILU(J) .NE. 1) GOTO 2
S = S + CKT(J,1) * A1(J,I)
2 CONTINUE
DIRECT(I) = C1(I) - S
1 CONTINUE
RETURN
END
```

X : tableau définissant le point considéré $[X^r]$

DIRECT : tableau de N éléments définissant la direction de montée.

N, NA, IR : définis comme dans REMAX.

IVL : défini comme dans XMAX.

ILU : tableau unidimensionnel (définissant la nature des contraintes)

$ILU_i = 1$ si la ième contrainte est saturée

$ILU_i = 0$ sinon.

- e) Sous-programmes QUADRA calculant la direction de montée des variables libres dans le cas où la méthode de quadratisation partielle est utilisée.

```

1:      SUBROUTINE DIRMEN(X,X0,A,B,NA,N,INDEX,P1,IVL,KRE)
2:      DIMENSION X(1),X0(1),A(NA,1),B(NA,1),P1(1),IVL(1),KRE(1)
3:      CALL XMAB(X0,P1,N,A,B,IVL,NA)
4:      DO 1 I=1,N
5:      DO 1 J=1,N
6:      1 A(I,J)=-A(I,J)
7:      L=1
8:      34 IF(A(L,L)) 30,30,31
9:      30 DO 32 M=1,N
10:     A(L,M)=0.
11:     32 A(M,L)=0.
12:     KRE(L)=0
13:     L=L+1
14:     IF(L-N) 100,100,101
15:     100 GOTO 34
16:     31 A(L,L)=DSQRT(A(L,L))
17:     KRE(L)=1
18:     IF(L-N) 200,71,71
19:     200 DO 21 J =L+1,N
20:     21 A(J,L)=A(J,L)/A(L,L)
21:     DO 10 J =L+1,N
22:     S=0.
23:     DO 23 I=1,J-1
24:     23 S=S+A(J,I)*A(J,I)
25:     S=A(J,J)-S
26:     IF(S)5,5,6
27:     5 KRE(J)=0
28:     DO 4 K=1,J
29:     4 A(J,K)=0.
30:     IF(J-N)70,71,71
31:     70 DO 66 K=J+1,N
32:     66 A(K,J)=0.
33:     GOTO 10
34:     6 KRE(J)=1
35:     A(J,J)=DSQRT(S)
36:     IF(J-N)80,71,71
37:     80 DO 8 I=J+1,N
38:     S=0.
39:     DO 8 K=1,J-1
40:     8 S=S+A(I,K)*A(I,K)
41:     S=A(I,J)-S
42:     27 A(I,J)=S/A(J,J)
43:     10 CONTINUE
44:
45:     71 IG=1
46:     42 IF(      KRE(IG)) 40,41,40
47:     41 X(IG)=0.,B(IG,1)=0.,IG=IG+1
48:     GOTO 42
49:     40 X(IG)=B(IG,1)/A(IG,IG)
50:     B(IG,1)=X(IG)
51:     IF(IG=N)90,91,91

```

```

20 20 17 I = IG+1, N
    IF( KRE(I)) 15,16,15
15 X(I)=0.
    B(I,1)=0.
    GOTO 17
16 TX=0.
    DO 12 J=1, I-1
12 TX=TX+A(I,J)*X(J)
    X(I)=(B(I,1)-TX)/A(I,I)
    B(I,1)=X(I)
17 CONTINUE
91 JG=N
22 IF( KRE(JG)) 50,51,50
51 X(JG)=0., B(JG,1)=0., JG=JG-1
    GOTO 52
50 X(JG)=B(JG,1)/A(JG,JG)
    I=(JG-1) 60,1001,60
60 DO 20 K=N-JG+2, N
    I=(N+1)-K
    IF( KRE(I)) 18,19,18
19 X(I)=0.
    B(I,1)=0.
    GOTO 20
18 TX=0.
    DO 14 L=I+1, N
    J=N+(I+1)-L
14 TX=TX+A(J,I)*X(J)
    X(I)=(B(I,1)-TX)/A(I,I)
20 CONTINUE
    INDI=0
    GOTO 1001
101 INDI=1
1001 RETURN
    END

```

X0 : tableau définissant le point considéré $[X^r]$

X : tableau définissant la direction de montée

INDEX : indicateur qui vaut 0 si le calcul de X est possible et vaut 1 dans le cas contraire

N, NA, KRE définis comme dans REMAX

A : tableau bidimensionnel de NA lignes et de NA colonnes

B : tableau bidimensionnel de NA lignes et d'une colonne

P1, IVL : tableau unidimensionnel.

5) Exemple d'utilisation

Nous prenons l'exemple du problème de contrôle optimal.

JOB: B-MALEGR15-ABC, IUCA, NGUYEN
 LIMIT (TIME, 7), (CORE, 44), (SPDISC, 40), (PAGES, 150)
 FORTRAN SILES, G8, ADP, NS

FUNCTION G(XA)

DIMENSION XA(11)

DIMENSION X0(5), X(5), C(5), XK(5), XK0(5), E(5), Z(5)

C(1) = 1.02 ; E(1) = 15000.

C(3) = 0.386 ; E(3) = 15000.

C(2) = 0.93 ; E(2) = 14000.

C(4) = 3.28 ; E(4) = 10000.

C(5) = 0.084 ; E(5) = 15000.

NV = 3

NM = 5

TF = 0.50 ; TO = 0.0 ; PT = 0.01

K = 50

R = 1.9865

X0(1) = 1.0 ; X0(2) = X0(3) = 0.0

X0(1) = 1.0 ; X0(2) = X0(3) = 0.0

T=TO

CALCULER DES XK0(I) = COEFFICIENTS K A L'INSTANT T=0.0

TE0 = XA(1)*DEXP(-XA(2)*T)

D9 11 I = 1, NM

Z(I) = -E(I)*(1./TE0-1./658.)/R

11 XK0(I) = C(I)*DEXP(Z(I))

D9 8 IPP = 1, K

T=T+PT

TE = XA(1)*DEXP(-XA(2)*T)

D9 1 I = 1, NM

Z(I) = -E(I)*(1./TE-1./658.)/R

1 XK(I) = C(I)*DEXP(Z(I))

SK=XK(1)+XK(2)+XK(3)

SK0 = XK0(1)+XK0(2)+XK0(3)

X(1) = X0(1)*(2.+PT*SK0)/(2.+PT*SK)

X4=2.+PT*XK(4)

X(2) = (X0(2)*(2.+PT*XK0(4)))/X4

1 +(X0(1)*PT*XK0(1))/X4

1 +(X(1)*PT*XK(1))/X4

XV=2.+PT*XK(5)

X(3) = (X0(3)*(2.+PT*XK0(5)))/XV

1 +(X0(2)*PT*XK0(4))/XV

1 +(X(2)*PT*XK(4))/XV

D9 4 I = 1, NV

4 X0(I)=X(I)

D9 5 I = 1, NM

5 XK0(I) = XK(I)

8 CONTINUE

G = X(3)

RETURN

```

SUBROUTINE DEPCOSAL(XA,A,IVL,ILU,NA)
DIMENSION XA(1),A(NA,1),IVL(1),ILU(1)
IF(IVL(1).NE.1) GOTO 1
A(1,1) = -1. ; A(2,1) = 1. ; A(3,1) = 0.0
1 IF(IVL(2).NE.1) GOTO 2
A(1,2) = 0.0 ; A(2,2) = 0.0 ; A(3,2) = 1.0
2 RETURN
END

```

```

SUBROUTINE CSSALYAI(XA,C,ILU)
DIMENSION XA(1),C(1),ILU(1)
IF(ILU(1).NE.1) GOTO 1
C(1) = 324.0 - XA(1)
1 IF(ILU(2).NE.1) GOTO 2
C(2) = XA(1) - 273.00
2 IF(ILU(3).NE.1) GOTO 3
C(3) = XA(2)
3 RETURN
END

```

```

SUBROUTINE GRADSF(XA,C1)
DIMENSION XA(1),C1(1)
PH = 1.0E-03

```

```

F1 = G(XA)
XA(1) = XA(1) + PH
F2 = G(XA)
XA(1) = XA(1) - PH
XA(2) = XA(2) + PH
F3 = G(XA)
XA(2) = XA(2) - PH

```

```

C1(1) = (F2-F1)/PH
C1(2) = (F3-F1)/PH

```

```

RETURN
END

```



```

DIMENSION XO(10),PO(10),X1(10),DI(10),CKT(10,1),A1(10,10),B1(10,
1),A(10,10),B(10,10),C1(10),F1(10),H1(10),KRE(10)
DIMENSION IVL(10),ILU(10),IVX(10),IP1(10),AL(10),Y(10),SUM(10)
COMMON/NGN/IR,LP,N,NA,MAX,NPAMAX,NBMAX,NTEMAX,IVL,ILU,IVX,IP1,
1/XGN/TEO,EPS,EPS1,EPS2,EPS3,AL,Y,SUM
NA=10;N=2;LP=0;IR=3;TEO=1;NPAMAX=10;NTEMAX=20;MAX=100;NBMAX=20
EPS=1.0E-03;EPS1=1.0E-15;EPS2=1.0E-02;EPS3=1.0E-03
READ(105,500)(XO(II),II=1,N)
READ(105,500)(PO(II),II=1,N)
500 FORMAT(F10.6,F10.6,F10.6,F10.6,F10.6,F10.6)
WRITE(108,102)
102 FORMAT(1H0,3X,'POINT INITIAL'/)
OUTPUT,(XO(I),I=1,N)
WRITE(108,103)
103 FORMAT(1H0,3X,'PAS INITIAL'/)
OUTPUT,(PO(II),II=1,N)
DO 2 I=1,IR
  IVX(II)=0
  CALL GJACNT(XO,PO,X1,FM,INDI,DI,CKT,A1,B1,A,B,C1,p1,H1,XM)
  WRITE(108,106)
106 FORMAT(1H0,3X,'VALEUR DE L INDICATEUR INDI'/)
  OUTPUT,INDI
  WRITE(108,104)
104 FORMAT(1H0,3X,'POINT OPTIMUN'/)
  OUTPUT,(X1(I),I=1,N)
  WRITE(108,105)
105 FORMAT(1H0,3X,'VALEUR DE LA FONCTION AU POINT OPTIMUN'/)
  OUTPUT,FM
  STOP
END
ASSIGN ET1,FIL,((STS,8LU)),(NAM,F4LIB),(UNT,AC,1SYS)
LINK
OPTION (UNSAT,ET1)
RUN

```

Résultats :

0 POINT INITIAL

XO(I) = 800.0000000000000

XO(I) = .5000000000000000

0 PAS INITIAL

PO(I) = .10000000000000000

PO(I) = .10000000000000000

0 VALEUR DE L INDICATEUR INDI

INDI = 0

0 POINT OPTIMUN

X1(I) = 823.0000000000000

X1(I) = .392697238043251

0 VALEUR DE LA FONCTION AU POINT OPTIMUN

FM = .425523941322983



BIBLIOGRAPHIE

- I BILIOUS O. - AMUNDSON N.R.
Optimisation température gradient in tabular reactors,
Chemical Engineering Science, 5, 81, 115 (1956)
- II CHAU NE RKASEN
Optimisation par une méthode du gradient avec accélération de convergence - Problèmes avec contraintes
Thèse 3ème Cycle - Université de Nancy I (1972)
- III KOVALIK J - OSBORNE M.R.
Methods for unconstrained optimisation problems
Ed. American Elsevier Publishing Company
Inc New-York
- IV LEGRAS J.
Cours du D.E.A. professé à l'U.E.R. Sciences
Mathématiques de l'Université de Nancy I (1972 - 1973)
- V LEGRAS J.
Méthodes et techniques d'analyse numérique
Ed. Dunod Paris (1971)
- VI PALLU DE LA BARRIERE
Cours d'automatique théorique
Ed. Dunod (1966)
- VII ROSENBROCK H.H. - STOREY C.
Computing methods in optimization problems
A.P. New-York, London (1964)
- VIII VIENNEY R.
Optimisation avec contraintes - Essai de la méthode des directions conjuguées.
Thèse 3ème cycle - Université de Nancy I (1968)
- IX WILLARD I. Z.
Nonlinear programming
Prentice Hall, Inc., Englewood Cliffs, N.J. (1969)

NOM DE L'ETUDIANT : NGUYEN VAN TUAN

NATURE DE LA THESE : DOCTEUR INGENIEUR

Numéro du C.N.R.S. : A.O 8655

VU, APPROUVE

& PERMIS D'IMPRIMER

NANCY, le 4 juin 1973 .

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

J.R. HELIOT



